PRONUNCIATION VARIANTS GENERATION USING SMT-INSPIRED APPROACHES

Panagiota Karanasou and Lori Lamel

Spoken Language Processing Group, LIMSI-CNRS
91403 Orsay, FRANCE
{pkaran,lamel}@limsi.fr

ABSTRACT

Enriching a pronunciation dictionary with phonological variation is a challenging task, not yet solved despite several decades of research, in particular for speech-to-text transcription of real world data where it is important to cover different pronunciation variants. This paper proposes two alternative methods, inspired by machine translation, to derive pronunciation variants from an initial lexicon with limited variations. In the first case, an n-best pronunciation list is extracted directly from a machine translation tool, used as a grapheme-to-phoneme (g2p) converter. The second is a novel method based on a pivot approach, previously used for the paraphrase extraction task, and here applied as a post-processing step to the g2p converter. Some preliminary speech recognition experiments with the automatically generated pronunciation variants are reported using Quaero development data.

Index Terms— pronunciation variants, SMT, pivot paraphrasing, g2p conversion, English

1. INTRODUCTION

Finding the pronunciation of a word from its written form (g2p conversion), has long been a research topic and has many applications, especially in speech synthesis and recognition, and spelling checkers. For many applications it is important to predict pronunciation of new terms or to add alternative pronunciations for existing ones.This is a difficult problem since the pronunciation of a given word depends on a number of diverse factors such as the linguistic origin of the word and the speaker, the speaker's education level, and the conversational context. It is widely acknowledged that the pronunciation dictionary contributes to the overall performance of automatic speech recognition (ASR) and synthesis systems.

Historically a substantial amount of manual effort is involved in the creation of a pronunciation lexicon. However, with the large vocabularies used in automatic systems, there has been a move towards data-driven approaches, based on the idea that given enough examples it should be possible to predict the pronunciation of an unseen word simply by analogy. Some proposed machine learning techniques are neural networks [1], decision trees [2] and hidden Markov models [3]. Joint-sequence models [4], [5] were shown to achieve better performance by pairing letter substrings with phoneme substrings. Recently, g2p conversion has been seen as a statistical machine translation (SMT) problem. Moses, a publicly available phrase-based SMT toolkit [6], was used for g2p conversion and tested in French [7] and Italian [8] ASR.

This work aims to generate pronunciation variants in an automatic and language independent way, even when no variants are included in the lexical resources available for training. Most available dictionaries contain no or few variants, or their variants are not consistent or suitable for training. The proposed methods are tested for English, a language known to be difficult for pronunciation generation. In [9], Moses was used as a phoneme-to-phoneme (p2p) converter to generate variants from baseform pronunciations when variants are included in the training set. However, when a single-pronunciation dictionary is used for training, it fails. In this latter case a novel approach, originally used for the paraphrase extraction task, was proposed to use also graphemic information to generate variants. This novel approach is based on the principle that sequences of modified phonemes can be identified using a graphemic sequence as a pivot. In [9], this pivot-based method was used to generate variants given a canonical pronunciation of a word. Here the method is used as a post-processing step to a g2p converter, enabling the generation of pronunciations with variants for out-of-vocabulary words (OOVs). It is independent of the origin of the input pronunciations, focusing on local variations, which are the most common pronunciation variants. The Moses toolkit is used as a g2p converter in this work, which provides an alternative method to derive variants via n-best lists.

2. METHODOLOGY

2.1. Generation of n-best lists by Moses

When Moses is used for g2p conversion, a pronunciation dictionary is used in the place of an aligned bilingual text corpora. The orthographic transcription is considered as the source language and the pronunciation as the target language.

This work is partly realized as part of the Quaero Programme (www.quaero.org), funded by OSEO, the French State agency for innovation and by the ANR EdyLex project.

This method has the desired properties of a g2p system: To predict a phoneme from a grapheme, it takes into account the local context of the input word and of the output pronunciation from a phrase-based model and allows sub-strings (phrases) of graphemes to generate phonemes. A 5-gram phoneme language model (LM), estimated on the pronunciations in the training set using the SRI toolkit [10], is used to provide additional phonemic information and corresponds to the target LM in SMT. Finally, the combination of all components is fully optimized with a minimum error training step (tuning) on a development set. The tuning strategy used was the standard Moses training framework based on the maximization of the BLEU score.

Moses can output an n-best translation list, a ranked list of translations of a source string. The 1-, 2-, 5- or 10-best translations (i.e. pronunciation variants) per word are kept.

2.2. Pivot paraphrasing approach

The pivot method applies the work of [11] to the generation of pronunciation variants. Paraphrases are alternative ways of conveying the same information. The analogy with pronunciation variants of a word is easily seen: the different pronunciations being alternate phonemic expressions of the same orthographic information. The paraphrases are phonemic phrases of the phrase table generated by Moses from the word-pronunciation training pairs. For each phonemic phrase in the translation table, we find all corresponding graphemic phrases and then look back to find what other phonemic phrases are associated with the set of graphemic ones. These phonemic phrases are plausible paraphrases.

In the following, f is a graphemic phrase and e_1 and e_2 phonemic phrases. The paraphrase probability $p(e_2 \mid e_1)$ is assigned in terms of the translation phrase table probabilities $\phi(f \mid e_1)$ and $\phi(e_2 \mid f)$ estimated based on the counts of the aligned graphemic-phonemic phrases. Since $e1$ can be translated as multiple graphemic phrases, we sum over f for all the graphemic entries of the phrase translation table:

$$\hat{e}_2 = \arg\max_{e_2 \neq e_1} p(e_2 \mid e_1) \quad (1)$$

$$= \arg\max_{e_2 \neq e_1} \sum_f \phi(f \mid e_1)\phi(e_2 \mid f) \quad (2)$$

An example of a paraphrase pattern in the dictionary is:

discou**nt**ed	diskW**nt**xd	dIskW**n**xd
discou**nt**enance	dIskW**nt**Nxns	dIskW**n**Nxns

The alternative pronunciations differ only in the part that can be realized as either **nt** or **n**, while the rest remains the same. The **nt** and **n** form a paraphrased pair. The pivot method focuses on local modifications observed between variants of a word and is a lot faster than the n-best list generation by Moses-g2p. All occurrences of these paraphrased patterns are substituted in the input pronunciations (the 1-best pronunciations of Moses-g2p).

At this point, different types of pruning are applied on the generated variants. First, the candidate variants are reranked based on additional phonemic contextual information expressed by 5-gram phoneme LM already used by Moses for the g2p conversion. The SRI toolkit served for the reranking. Then, pruning is done based on the length of the paraphrases substituted in the pronunciations. It was experimentally found that the quality of the generated variants improves when only 3- and 4-grams paraphrases are substituted because more context is taken into account throughout the procedure and some confusions are avoided.

The Levenshtein Distance between each pronunciation and its generated variants was then calculated. This measure should not exceed a threshold since the different pronunciations of a word are usually phonemically very close. Pruning with thresholds of 3 (LD3) and 2 (LD2), meaning that all the variants with edit distances greater than 3 and 2 respectively are pruned, were tried. Finally, the 1-, 4- and 9-best pronunciation variants per input pronunciation were kept and merged with the input pronunciations in order to have 2-, 5- and 10-best pronunciations generated and so as to be able to compare these with the n-best lists from Moses g2p.

3. EXPERIMENTAL SETUP

The LIMSI American English pronunciation dictionary, created with extensive manual supervision, serves as basis of this work. The pronunciations are represented using a set of 45 phonemes [12]. 18% of the words are associated with multiple pronunciations. These mainly correspond to well-known phonemic alternatives (for example the pronunciation of the ending "ization"), and to different parts of speech (noun or verb). The dictionary contains a mix of common words, acronyms and proper names, the last two categories being difficult cases for g2p converters.

Only the canonical pronunciation of each word was used to train the g2p converter. Since canonical pronunciations are not explicitly indicated in the lexicon, the longest one is taken as the canonical form. The dictionary was randomly split into a training, a development (dev) and a test set. There are 160k distinct entries (word-pronunciation pairs) in the training set, 9k distinct entries in the dev set and 16k entries (words with one or multiple pronunciations) in the test set.

4. EVALUATION

Recall and phone error rate (PER) are used to evaluate the predictions of one or multiple pronunciations. Word x_i of the test set (i=1..w) has j distinct pronunciations y_{ij} (y_i is a set with elements $y_{ij}, j = 1..d_i$). Our systems can generate one or more pronunciations $f(x_i)$ ($f(x_i)$ is also a set). Recall (R) is conventionally defined and calculated on all references (canonical pronunciations and variants) to evaluate the g2p conversion. It is also computed only on the variants in order to specifically evaluate their correctness. [1] The PER is measured

[1] Macro-recall, defined in [9], gives more weight to examples with multiple variants. Macro-recall gave similar results to the conventional recall.

using the Levenshtein Distance (LD) between the generated pronunciations and the reference pronunciations:

$$PER_{n-best} = \frac{\sum_{i=1}^{w} \min LD(y_i, f(x_i))}{\sum_{i=1}^{w} |y_i|} \quad (3)$$

$$PER_{1-best} = \frac{\sum_{i=1}^{w} \min LD(y_i, f(x_i))}{\sum_{i=1}^{w} |y_{im}|} \quad (4)$$

where y_{im} is the pronunciation of the word x_i with a minimum LD.

The Moses-g2p converter (M-g2p) and the pivot paraphrasing method (P) were tested with single-pronunciation training. The PER on the test set is presented in Table 1 for Moses-g2p and Pivot with LD2 pruning (P LD2). The PER is about 6% for the 1-best Moses-g2p pronunciation, and 1.26% if the 10-best pronunciations are considered. The string error rate (SER) is 25%. Since the 1-best pronunciations generated by Moses-g2p are used as input to the pivot post-processing, the corresponding entry in the table is empty for Pivot.

Table 2 gives recall results compared to all references (top) and only variants (bottom) with both methods. Precision was also calculated, but only recall is presented because we consider it more important to cover possible pronunciations than to have too many, since other methods can be applied to reduce the overgeneration (alignment with audio, manual selection, use of pronunciation probabilities, etc). The best value that both precision and recall can obtain is 1. In terms of recall measured on all references (R-all ref), it can be seen that Moses-g2p outperforms the pivot-based method (with and without LD pruning). The best results is a recall of 0.91 when using the 10-best pronunciations generated by Moses g2p.

It should be pointed out that the all reference measures (recall and PER) favor the Moses-based approach because the pivot-based approach aims at generating variants. This is why we also evaluated the recall only on variants as given in the lower part of Table 2. For the variants-only case (R-variants) it can be seen in that pivot with LD2 or LD3 pruning outperforms Moses-g2p. It manages to generate more correct variants when no variants are given in the training set. Pivot takes directly the variation patterns from the phrase table of Moses avoiding the overfitting effects of the EM algorithm used by Moses for the construction of a generative model. Moreover, to reduce the overall complexity of decoding, the search space of Moses is typically pruned using simple heuristics and, as a consequence, the best hypothesis returned by the decoder is not always the one with the highest score. We plan to experimentally verify this theoretical error analysis in future work.

Table 1. *PER on all references (canonical pron+variants) for Moses-g2p (M-g2p) and Pivot (P)*

Method	Measure	1-best	2-best	5-best	10-best
M-g2p	PER (%)	**6.22**	3.99	1.98	**1.26**
P LD2	PER (%)	-	6.17	5.16	3.52

Table 2. *Recall on all references (canonical pron+variants) and only on variants for Moses-g2p (M-g2p) and Pivot (P)*

Method	Measure	1-best	2-best	5-best	10-best
M-g2p	R-all ref	0.68	0.79	0.88	**0.91**
P	R-all ref	-	0.72	0.78	0.82
P LD3	R-all ref	-	0.72	0.78	0.82
P LD2	R-all ref	-	0.72	0.78	0.83
M-g2p	R-variants	0.10	0.25	0.44	**0.55**
P	R-variants	-	0.19	0.32	0.44
P LD3	R-variants	-	0.35	0.49	0.60
P LD2	R-variants	-	0.36	0.50	**0.61**

Last but not not least, the reference dictionary is mostly manually constructed and certainly incomplete with respect to coverage of pronunciation variants particularly for uncommon words. This means that some of the generated variants are likely to be correct (or plausible) even if they are not in the references used in this evaluation.

5. SPEECH RECOGNITION EXPERIMENTS

The pronunciations generated by the Moses-g2p were further tested in two preliminary speech recognition experiments. In the first, automatically generated variants are added to a single-pronunciation dictionary, and in the second we simulate adding pronunciations for OOV words. To our knowledge it is the first time they are tested in a state-of-the-art ASR for English broadcast data. The speech transcription system uses the same basic modeling and decoding strategy as in the LIMSI English broadcast news system [13].

The acoustic models (AMs) are gender-dependent, speaker-adapted, and Maximum Likelihood trained on about 500 hours of audio data. They cover about 30k phone contexts with 11600 tied states. N-gram LMs were trained on a corpus of 1.2 billion words of texts from various LDC corpora (English Gigaword, BN transcriptions, commercial transcripts), news articles downloaded from the web, and assorted audio transcriptions. The recognition word list contains 78k words, selected by interpolation of unigram LMs trained on different text subsets as to minimize the OOV rate on a set of independent development texts. Word recognition was performed in a single real-time decoding pass, generating a word lattice with cross-word, position-dependent AMs, followed by consensus decoding [14] with a 4-gram LM. Unsupervised AM adaptation is performed for each segment cluster using the CMLLR and MLLR techniques prior to decoding.

The Quaero 2010 development data were used in these experiments. This 3.5 hour data set contains 9 audio files recorded in May 2010, covering a range of styles, from broadcast news (BN) to talk shows. Roughly 50% of the data can be classed as BN and 50% broadcast conversation (BC). These data are considerably more difficult than pure BN data. The overall word error rate (WER) with the original recognition dictionary is 30%, but the individual shows vary from 20% to over 40%. These are competitive WERs on these data.

In Table 3, the n-best pronunciations (1-, 2- and 5-best)

Table 3. *WER(%) adding Moses nbest-lists (M1, M2, M5) to single pronunciation baselines.*

System	Baseline	M1	M2	M5
Baseline longest	41.6	38.2	38.4	40.8
Baseline most frequent	32.9	32.0	33.4	37.3

Table 4. *WER(%) generating prons for OOVs using l2s and Moses nbest-lists (M1, M2, M5, M10).*

l2s	M1	M2	M5	M10
37.8	31.3	31.2	32.0	33.2

generated by the Moses-based system under the single-pronunciation training condition, are added to the canonical pronunciation of the original recognition dictionary (Baseline longest). Then, the same pronunciations are added to the most frequent one (Baseline most frequent). The results show that using only the longest pronunciation results in a large increase in WER. Adding pronunciations improves over the baseline longest dictionary, up until the 5-best pronunciations. The most frequent pronunciation baseline dictionary has a WER closer to the baseline of the original multiple pronunciation dictionary. In this case adding one pronunciation improves the performance of the ASR system, but adding more pronunciations degrades it. We expect that using pronunciation probabilities can reduce this degradation.

We then simulated the generation of pronunciations for OOVs. Starting with the full dictionary with variants, the pronunciations for the 20% least frequent words in the test data (about 7% of dictionary) were replaced with automatically generated pronunciations using the Moses g2p system. For comparison, the g2p system l2s [15], obtained from the Cambridge University ftp server and slightly modified to use the phone set of the LIMSI ASR system, was used to generate pronunciations for the missing words. [12] reports that this system provided consistent pronunciations and gave satisfactory recognition results. The results in Table 4 show that the dictionary with Moses-based pronunciations for the OOVs outperforms those of the l2s system, even with 10 pronunciations, even though these many alternatives can be expected to cause confusions. We add up to ten pronunciations because in Table 1 it is shown that there is a significant improvement to the PER passing from 5-best to 10-best generated pronunciations (36% relative) and thus in the quality of the generated pronunciations.

Nevertheless, in neither case was the performance of the original multiple pronunciation dictionary achieved. This dictionary is a difficult baseline because it is mostly manually constructed and well-suited to the needs of an ASR system. However, we expect that it is possible to obtain additional gains if probabilities are added to the generated pronunciation variants to moderate confusability.

6. CONCLUSION AND DISCUSSION

This paper has reported on generating pronunciation variants by a proposed novel pivot-based method and compared this approach with directly taking the n-best lists from a Moses SMT system. The n-best lists of Moses have a higher recall when a comparison is made to all reference pronunciations in the test set. However, the pivot-based method generates more correct variants. This is an advantage of the pivot method that could be useful in certain cases, for example to generate variants from the output of a rule-based g2p system which, if originally developed for speech synthesis, may not model pronunciation variants or to enrich a dictionary with limited pronunciation variants.

The pronunciations generated by Moses were also used to carry out preliminary tests in a state-of-the-art ASR system. These experiments show that the added pronunciations are of good quality even though trained under limited variation conditions and can improve the single pronunciation baselines. Our point in this paper is not, however, to present an ASR system and focus on the improvement of its performance, but to propose data-based approaches for variant generation that better model variability in spoken language. In the future, we plan to further evaluate the pronunciations generated by pivot by measuring their influence in ASR systems for different data sets (broadcast news, conversational speech). A continuation of the present work will be to find a way to add probabilities to the generated pronunciations to avoid confusion that may arise from adding a large number of variants.

7. REFERENCES

[1] T. Sejnowski and C. Rosenberg, "Nettalk: a parallel network that learns to read aloud," Report JHU/EECS-86/01, 1986.

[2] T.G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *J. Artificial Intelligence*, 1995.

[3] P. Taylor, "Hidden markov models for grapheme to phoneme conversion," Interspeech, 2005.

[4] M. Bisani and H. Ney, "Investigations on joint-multigram models for grapheme-to-phoneme conversion," ICSLP, 2002.

[5] S. Deligne, F. Yvon, and F. Bimbot, "Variable-length sequence matching for phonetic transcription using joint multigrams," Eurospeech, 1995.

[6] P. Koehn et al., "Moses:open source toolkit for statistical machine translation," ACL, 2007.

[7] A. Laurent, P. Deleglise, and S. Meignier, "Grapheme to phoneme conversion using an smt system," Interspeech, 2009.

[8] M. Gerosa and M. Federico, "Coping with out-of-vocabulary words:open versus huge vocabulary ASR," ICASSP, 2009.

[9] P. Karanasou and L. Lamel, "Comparing smt methods for automatic generation of pronunciation variants," *IceTAL*, 2010.

[10] A. Stolcke, "Srilm-an extensible language modeling toolkit," ICSLP, 2002.

[11] C. Bannard and C. Callison-Burch, "Paraphrasing with bilingual parallel corpora," ACL, 2005.

[12] L. Lamel and G. Adda, "On designing pronunciation lexicons for large vocabulary, continuous speech recognition," ICSLP, 1996.

[13] J.L. Gauvain, L. Lamel, and G. Adda, "The limsi broadcast news transcription system," *Speech Communication*, 2002.

[14] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: Lattice-based word error minimization," Eurospeech, 1999.

[15] J.A. Wasser, "English to phoneme translation, final version (4/15/85)," Cambridge Univ: svr-ftp.eng.cam.ac.uk.