AUTOMATIC GENERATION OF A PRONUNCIATION DICTIONARY WITH RICH VARIATION COVERAGE USING SMT METHODS

Panagiota Karanasou and Lori Lamel

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

Abstract. Constructing a pronunciation lexicon with variants in a fully automatic and language-independent way is a challenging issue, with many applications in human language technologies. Moreover, with the growing use of web data, there is a recurrent need to add words to existing pronunciation lexicons, and an automatic method can greatly simplify the effort required to generate pronunciations for these out-of-vocabulary words. In this paper, a machine translation approach is used to perform grapheme-to-phoneme (g2p) conversion, the task of finding the pronunciation of a word from its written form. Two alternative methods are proposed to derive pronunciation variants. In the first case, an n-best pronunciation list is extracted directly from the g2p converter. The second is a novel method based on a pivot approach, traditionally used for the paraphrase extraction task, and applied as a post-processing step to the g2p converter. The performance of these two methods is compared under different training conditions. The range of applications which require pronunciation lexicons is discussed and the generated pronunciations are further tested with some preliminary experiments in automatic speech recognition.

Key words: pronunciation lexicon,G2P conversion, SMT, pivot paraphrasing

1 Introduction

Grapheme-to-phoneme conversion (g2p) is the task of finding the pronunciation of a word given its written form. Despite several decades of research, it remains a challenging task with many applications in human language technologies. Predicting pronunciations and variants, that is, alternative pronunciations observed for a linguistically identical word, which cover all possible cases is a complicated problem that depends on a number of diverse factors such as the linguistic origin of the speaker and of the word, the education and the socio-economic level of the speaker and the conversational context.

Several approaches have been proposed in the literature to generate pronunciations. The simplest technique is dictionary look-up, but making a pronunciation dictionary by hand requires specific linguistic skills and necessarily has limited coverage. Rule-based conversion systems, which have a predominantly one-to-one correspondences between letters and predicted phonemes, still require specific linguistic knowledge and do not always capture the irregularities of a natural language even if exception rules or lists are included.

In contrast to knowledge-based approaches, data-driven approaches are based on the idea that given enough examples it should be possible to predict the pronunciation of an unseen word simply by analogy. A variety of machine learning techniques have been applied to this problem in the past including neural networks [15] and decision trees [4] that predict a phoneme for each input letter using the letter and its context as features, but do not consider -or consider very limited- context in the output. Other techniques allow previously predicted phonemes to inform future decisions such as HMM in [17] but they do not take into account the input's letter context. Joint-sequence models have been proposed [2], [3], that achieve better performance by pairing letter substrings with phoneme substrings, allowing context to be captured implicitly by these groupings. Other methods using many-to-many correspondences, as the one proposed in [7] report high accuracy.

Another machine learning approach that has been tried recently is to envisage the problem of g2p conversion as a statistical machine translation (SMT) problem. Moses, a publicly available phrase-based statistical machine translation toolkit [9], has been used for g2p conversion of French [11] and Italian [6] and other languages [13]. In this work, the aim is to generate pronunciations with variants for the English language. It should be noted that English is a difficult language for g2p conversion, since there is a loose relationship between letters and sounds. In a first step, Moses is used as a g2p converter. Then, two options are explored to generate pronunciation variants. In the first one, variants are derived from the generated n-best list. The second method is based on the idea that paraphrases in one language can be identified using a phrase in another language as a pivot. In the case of multiple pronunciation generation, sequence s of modified phonemes found in the pronunciation variants are identified using a sequence of graphemes in the corresponding word as a pivot. This method can also be used independently to generate alternative pronunciations from a canonical pronunciation of a word, thereby enriching the dictionary. Here it is used as a post-processing step to the g2p converter. It is an alternative to the direct generation of an n-best list by the g2p converter, independent of the origin of the input pronunciations, focusing on local variations, which are the most common variations found in multiple pronunciations of a word and permitting more generalization in the variants generation as will be explained later. To the best of our knowledge, this is the first application of a pivot approach to the generation of pronunciation variants.

The paper is organized as follows. Section 2 describes the two methods used in this study. Section 3 describes the experimental framework and details about the corpora used and the training conditions applied. Section 4 presents the evaluation results of the automatic generation of multiple pronunciations, while in Section 5 some applications of the generated dictionary are discussed and some preliminary speech recognition experiments are conducted to test the generated pronunciations in an applicative task. Conclusions and discussions for future work are reported in Section 6.

2 Methodology

This section first describes Moses as a g2p converter, and then presents two methods to generate variants. 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 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 of graphemes to generate phonemes. The phoneme sequence information is additionally modeled by a phoneme n-gram language model (LM) that corresponds to the target language model in machine translation. In this study, a phoneme-based 5-gram LM built on the pronunciations in the training set using the SRI toolkit [16] is used.

Moses also calculates distortion models, but this is not necessary as g2p conversion is a monotonic task. 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.

2.1 Generation of n-best lists by Moses

Moses can also output an n-best translation list. This list gives a ranked list of translations of a source string with the distortion, the translation and the language model weights, as well as an overall score for each translation. The 2-, 5- or 10-best translations (i.e. pronunciation variants) per word are kept. Some words have fewer possible variants, in which case all variants are taken.

2.2 Pivot paraphrasing approach

This is an alternative method based on [1] for the generation of pronunciation variants, added as a post-processing step to the Moses-g2p converter. Paraphrases are alternative ways of conveying the same information. The analogy with multiple pronunciations of the same word is easily seen, the different pronunciations being alternate phonemic expressions of the same orthographic information. In [1], a paraphrase probability is defined that allows paraphrases extracted from a bilingual parallel corpus to be ranked using translation probabilities, and then these are reranked taking the contextual information into account. For the problem of automatic pronunciation variant generation, this bilingual corpus corresponds to the corpus of word-pronunciation pairs already used by Moses for g2p conversion and the paraphrases are phonemic phrases extracted from the translation table in that task. 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 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:

$$\begin{aligned} \hat{e}_2 &= \arg\max_{e_2 \neq e_1} p(e_2 \mid e_1) && (1) \\ &= \arg\max_{e_2 \neq e_1} \sum_f \phi(f \mid e_1)\phi(e_2 \mid f) && (2) \end{aligned}$$

This returns the single best paraphrase, $\hat{e}_2$, irrespective of the context in which e_1 appears. The paraphrased pairs with their probabilities are extracted for the input pronunciations, which are the pronunciations generated by Moses for the words of the test set during the g2p conversion task. The 10-best paraphrases for each input phonemic phrase found in the translation table are extracted with a maximum extent of 4 phonemes. 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. The generation of variants with pivot 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 for all the possible combinations (only in the first occurrence, only in the second, in the first and the second, etc.), limiting to 3 the maximum number of occurrences of the same paraphrase in a pronunciation.

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 a simple language model trained on the correct pronunciations of the training set. This is the same phoneme-based 5-gram language model used by Moses for the g2p conversion. The SRI toolkit was used for the reranking. Then pruning based on the length of extracted paraphrases substituted in the pronunciations is realized. Many errors were observed due to substitutions of unigram paraphrases that the reranking did not manage to handle successfully. It was experimentally found that the quality of the generated variants improves when only 3- and 4-grams paraphrases are substituted. This is normal as 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 (1-best pronunciations of Moses-g2p) in order to have 2-, 5- and 10-best pronunciations generated and to be able to compare these results with the n-best lists of Moses-g2p.

3 Experimental setup

The LIMSI American English pronunciation dictionary serves as basis of this work. We decided to use this dictionary as it is reputed to be a high quality dictionary for speech recognition, which will be the domain of application of the proposed methods in Section 5.1[1]. The dictionary has been created with extensive manual supervision and has 187975 word entries. Each dictionary entry contains the orthographic form of a word and its pronunciations (one or more). The pronunciations are represented using a set of 45 phones [10]. 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). Case distinction is eliminated since in general it does not influence the word's pronunciation, the main exceptions being acronyms which may have both a spoken and spelled form, but these are quite rare. Some symbols in the graphemic form are not pronounced, such as the hyphen in compound words. The dictionary contains a mix of common words, acronyms and proper names, the last two categories being difficult cases for g2p converters.

The corpus was randomly split based on the graphemic form of the word into a training, a development (dev) and a test set. The dev set is necessary for the tuning of the Moses model. In order to have a format that resembles the aligned parallel texts used for training machine translation models, the dictionary is expanded so that each entry corresponds to a word-one pronunciation pair. The resulting dev and test sets have 11k and 19k distinct entries.

The g2p converter is trained for two conditions, on the entire training subset using all pronunciations for words with multiple ones or on the same word list but using only one (canonical) pronunciation per word. Since canonical pronunciations are not explicitly indicated in the lexicon, the longest one is taken as the canonical form. In the first training condition, there are 200k entries (distinct

[1] Although not publicly available, this dictionary is available by request. It has been used by numerous laboratories. SRI, Philips Aachen, ICSI and Cambridge University have reported improving the performance of their systems using this dictionary.

word-pronunciation pairs) in the training set with on average 1.2 pronunciations/word. In the second training condition, the training set has 160k entries with a single pronunciation per word.

4 Evaluation

In this study, precision and recall, first introduced in information retrieval [14], as well as 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 $y_{ij}, j = 1..d_i$). Moreover, our systems can generate one or more pronunciations $f(x_i)$ ($f(x_i)$ is also a set). Recall (R) is conventionally defined:

$$\mathrm{R} = \frac{1}{w}\sum_{i=1}^{w}\frac{|f(x_i)\cap y_i|}{|y_i|} \tag{3}$$

Precision (Pr) is defined analogously as the number of correct generated pronunciations divided by the total number of generated pronunciations. They are calculated on all references (canonical pronunciations and variants) to evaluate the g2p conversion, but also only on the variants in order to specifically evaluate their correctness. The PER is measured using the Levenshtein Distance (LD) between the generated pronunciations and the reference pronunciations:

$$PER_{n-best} = \frac{\sum_{i=1}^{w}\sum_{j=1}^{d_i}\min LD(y_{ij}, f(x_i))}{\sum_{i=1}^{w}\sum_{j=1}^{d_i}|y_{ij}|} \tag{4}$$

$$PER_{1-best} = \frac{\sum_{i=1}^{w}\sum_{j=1}^{d_i}\min LD(y_{ij}, f(x_i))}{\sum_{i=1}^{w}|y_{im}|} \tag{5}$$

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

The Moses-g2p converter (M-g2p) and the pivot paraphrasing method (P) were tested for the multiple pronunciation and single pronunciation training conditions. Table 1 gives recall results compared to all references (top) and only variants (middle), as well as PER (bottom) with both methods for multiple pronunciation training. 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) and on recall on variants (R-variants), it can be seen in Table 1 that Moses-g2p outperforms the pivot-based method. The best result is a recall on all references of 0.94 when using the 10-best pronunciations generated by Moses-g2p. The PER (bottom) is about 6% for the 1-best Moses-g2p pronunciation, and 1.17% if the 10-best pronunciations are considered. The string error rate (SER) is 25%. Since the

Table 1. *Recall and PER on all references (canonical prons+variants) and only on variants for Moses-g2p (M-g2p) and Pivot (P) for multiple pronunciation training.*

Method	Measure	1-best	2-best	5-best	10-best
M-g2p	R-all ref	**0.68**	0.82	0.91	**0.94**
P LD2	R-all ref	-	0.74	0.80	0.84
M-g2p	R-variants	0.27	0.63	0.82	0.89
P LD2	R-variants	-	0.50	0.66	0.73
M-g2p	PER (%)	**6.13**	4.00	1.97	**1.17**
P LD2	PER (%)	-	6.00	4.47	3.52

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

Method	Measure	1-best	2-best	5-best	10-best
M-g2p	R-all ref	0.68	0.79	0.88	0.91
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**

1-best pronunciations generated by Moses-g2p are used as input to the pivot post-processing, the corresponding entries in the table are empty for Pivot.

In Table 2 the recall results on all references (top) and only on variants (bottom) for single pronunciation training are shown. For the recall on variants, the results of pivot without LD pruning are presented (P) as well as the results with LD threshold 3 (P LD3) and with LD threshold 2 (P LD2) because a large improvement can be seen in the intermediate pruning steps. The PER is not reported in this table since it does not change significantly from the results in Table 1.

The recall on all references (R-all ref) in Table 2 compared to the results in the top of Table 1 degrades by an average 3% absolut, but the variant recall degrades severely. However, for the latter case it can be seen 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.

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 gener-

Table 3. *Recall on variants for generation of 1-, 4- and 9-best variants for Moses-g2p (M-g2p) and Pivot (P) for the test set with the entire dict training condition*

Method	Measure	1-best	4-best	9-best
M-g2p	R	0.35	0.55	0.62
P LD2	R	0.23	0.39	0.46
P correct entry LD2	R	0.39	0.65	**0.75**

ating variants. This is why we also evaluated the recall only on variants. However, while the pivot method gives better results than Moses-g2p to generate variants with the single pronunciation training condition, this is not the case for the multiple pronunciation training condition. We wanted to further investigate the cause of this degradation. When the pivot is used as a post-processing step to the Moses-g2p converter, its input is the output of Moses which has PER of 6%, low enough to be reliable, but the SER is 25% which can plausibly degrade the performance of pivot. To verify this hypothesis, the pivot method was applied to the correct canonical pronunciation of the test set and these results were compared to the previous results of 1-, 4- and 9-best variants generated by pivot as well as to the variants generated by Moses-g2p. In these experiments, wanting to see purely the influence of variants generated by pivot, we do not add to them the 1-best pronunciation generated by Moses as previously done. To enable the comparison of the two methods, the first pronunciation generated by Moses-g2p is considered the canonical one and removed from the n-best lists. The results in Table 3 on recall computed on variants in the reference set reveal that pivot, when applied to a correct input, not only outperforms itself applied to a 'noisy' input, but also the Moses-g2p method. This is an important assumption, as there are cases where no multiple pronunciation lexicons are available for training but the enrichment of a single pronunciation dictionary is desired in order to make it usable in different tasks, for example a recognition system for conversational speech.

All results presented in this section are calculated with the full 45-phone set. However, some exchanges are less important than others. If some errors, such as the confusion between syllable nasals and a schwa-nasal sequence (the substitution of "N" by "xn") are not taken into account, the overall recall improves by 1-2% absolute for both methods, and the PER is reduced by 0.1-0.2% for Moses-g2p and 0.3-0.4% for pivot.

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. The pronunciations of words of foreign origin (mostly proper names) may also be incomplete since their pronunciation depends highly on the speaker's knowledge of the language of origin. 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 the upper evaluation.

5 Applications

An automated pronunciation dictionary with variants presents a great span of applications. First of all, it is an essential element of speech recognition and speech synthesis systems. In fact, the construction of a good pronunciation dictionary is important to ensure acceptable automatic speech recognition performance [10]. Moreover with the wide use of real data there are words not yet included in a recognition dictionary (out-of-vocabulary words), for which a pronunciation rapidly and automatically generated is often required. Another domain of application of the phonetization task in natural language processing is the detection and correction of orthographic errors [18], while the strong relation between phonology and morphology is well known and studied with morphological phenomena of purely phonological origins or guided by phonological constraints, among other interactions [8]. Other applications include computer-aided pronunciation training (CART) and in general e-learning systems.

5.1 Speech Recognition Experiments

To further test the pronunciations generated by the Moses-g2p method in an application framework, some preliminary speech recognition experiments were conducted. Similar experiments have been reported for the Italian [6] and French [11] languages, but to our knowledge they have never been 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 [5]. The acoustic models 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 out-of-vocabulary (OOV) rate on set of development texts. Word recognition was performed in a single real-time decoding pass, generating a word lattice followed by consensus decoding [12] with a 4-gram LM. Unsupervised acoustic model adaptation is performed for each segment cluster using the CMLLR and MLLR techniques prior to decoding.

The Quaero (www.quaero.org) 2010 development data were used in the recognition experiments. This 3.5 hour data set contains 9 audio files recorded in May 2010, covering a range 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) is 30%, but the individual shows vary from 20% to over 40%. These are competitive WERs on these data.

In Table 4, the n-best pronunciations (1-, 2- and 5-best) generated by the Moses-based system under the two training conditions, are added to the canon-

Table 4. *WER(%) adding Moses nbest-lists (M1, M2,M5) to the longest pronunciation baseline*

Training condition	M1	M2	M5
Single pronunciation	38.2	38.4	40.8
Multiple pronunciations	37.9	38.2	39.1
Baseline longest	41.6		

ical pronunciation of the original recognition dictionary (Baseline longest). 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 pronunciations trained under the multiple pronunciation training condition improve more the WER compared with the pronunciations trained with the single pronunciation dictionary. This is because the formers are trained to better model the variants which correspond to the reduced forms, closer to the spoken language most of the times.

In Table 5, the same pronunciations (M1, M2, M5) are added to the most frequent pronunciation of the recognition dictionary (Baseline most frequent). 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 (trained on a single or multiple pronunciation dictionary) improves the performance of the ASR system, but adding more pronunciations degrades it.

Althought the quality of the pronuciations trained on a multiple pronunciation dictionary is higher, measured with recall on all references and on variants, they are submitted to the same confusability effects. What is more, when adding two or five pronunciations to the most frequent baseline, the system with pronunciations trained on a single pronunciation presents lower WERs. An explanation could be that the pronunciations trained under multiple pronunciations can better represent reduced forms and, thus, are closer to the most frequent baseline and easier to be confused. An example of the introduced confusability is that of the multiple pronunciation training outputs for the word *you*. They are the pronunciations */yu/* and */yc/* when the 2-best list is kept. The latter pronunciation (*/yc/*) is not generated under the single pronunciation training. */yc/* in the phrase *you are* is easily confused with */ycr/*, the pronunciation of *your*. Such frequent cases can be responsible for the degradation of the ASR sys-

Table 5. *WER(%) adding Moses nbest-lists (M1, M2,M5) to the most frequent pronunciation baseline*

Training condition	M1	M2	M5
Single pronunciation	32.0	33.4	37.3
Multiple pronunciations	32.0	34.5	38.9
Baseline most frequent	32.9		

tem with pronunciations trained on the multiple pronunciation dictionary when many alternatives are added.

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 a fully automatic and language independent generation of pronunciations using Moses, an open-source SMT tool, as a g2p converter and generating pronunciation variants taking directly the n-best lists of Moses or applying a novel pivot-based method. The n-best lists of Moses yield better recall results than the pivot-based method on all references. However, it was shown that the pivot-based method can generate more correct variants. This is an advantage of the pivot method that could be useful in certain cases, especially in the case of limited variation in the training set, 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 generated pronunciations were also evaluated in an applicative task. They were used to carry out tests in a state-of-the-art ASR system. These experiments show that Moses provides variants of good quality that even without any further pruning can improve the one 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 the generation of a pronunciation dictionary with variants. 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). Another problem that interests us is generating pronunciations specifically for named entities (proper names, geographical names,etc.), which are very often cases of out-of-v0cabulary words and their pronunciations rarely follow regular phonological rules.

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

References

1. Bannard, C., Callison-Burch, C.: Paraphrasing with bilingual parallel corpora. In: Proc. of ACL (2005)
2. Bisani, M., Ney, H.: Investigations on Joint-Multigram Models for Grapheme-to-Phoneme Conversion. In: ICSLP (2002), 105-108

3. Deligne, S., Yvon, F., Bimbot, F.: Variable-length sequence matching for phonetic transcription using joint multigrams. In: Proc. European Conf. On Speech Communication and Technology (1995), 2243-2246
4. Dietterich, T.G., Bakiri, G.: Solving Multiclass Learning Problems via Error-Correcting Output Codes. In: Journal of Artificial Intelligence, Vol. 2 (1995), 263-286
5. Gauvain, J.L., Lamel, L., Adda, G.: The LIMSI Broadcast News Transcription System. In: Speech Comm., Vol. 37 (2002), 89-108
6. Gerosa, M., Federico, M.: Coping with out-of-vocabulary words:open versus huge vocabulary ASR. In: ICASSP (2009)
7. Jiampojamarn, S., Cherry, C., Kondrak, G.: Joint processing and discriminative training for letter-to-phoneme conversion. In: Proc. Of ACL-HLT (2008), 905-913
8. Kaisse, E. M.: Word-Formation and Phonology. In: Handbook of Word-Formation, Studies in Natural Language and Linguistic Theory, Springer Netherlands, Vol.64 (2005), 25-47
9. Koehn, P. et al.: Moses:Open source toolkit for statistical machine translation. In: ICSLP (2002)
10. Lamel, L., Adda, G.: On designing pronunciation lexicons for large vocabulary, continuous speech recognition. In: Proc. ICSLP (1996), 6-9
11. Laurent, A., Deleglise, P., Meignier, S.: Grapheme to phoneme conversion using an SMT system. In: Interspeech (2009)
12. Mangu, L., Brill, E., Stolcke, A.: Finding Consensus Among Words: Lattice-Based Word Error Minimization. In: Eurospeech (1999), 495-498
13. Rama, T., Singh, A. K., Kolachina, S.: Modeling Letter-to-Phoneme Conversion as a Phrase Based Statistical Machine Translation Problem with Minimum Error Rate Training. In: Proc. NAACL-HLT: Student Research Workshop & Doctoral Consortium (2009), 90-95
14. Van Rijsbergen, C.J.: Information Retrieval, Butterworths, London, UK. (1979)
15. Sejnowski, T., Rosenberg, C.: NETtalk: a parallel network that learns to read aloud. In: Report JHU/EECS-86/01 (1986)
16. Stolcke, A.: SRILM-An extensible language modeling toolkit. In: Proc. ICSLP-02 (2002)
17. Taylor, P.: Hidden Markov models for grapheme to phoneme conversion. In: Interspeech (2005), 1973-1976
18. van Berkel, B., De Smedt, K.: Triphone analysis:a combined method for the correction of orthographical and typographical errors. In: Proc. of the Second Conf. on Applied Natural Language Processing (1988), 77-83