Comparing SMT Methods for Automatic Generation of Pronunciation Variants

Panagiota Karanasou and Lori Lamel

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

Abstract. Multiple-pronunciation dictionaries are often used by automatic speech recognition systems in order to account for different speaking styles. In this paper, two methods based on statistical machine translation (SMT) are used to generate multiple pronunciations from the canonical pronunciation of a word. In the first method, a machine translation tool is used to perform phoneme-to-phoneme (p2p) conversion and derive variants from a given canonical pronunciation. The second method is based on a pivot method proposed for the paraphrase extraction task. The two methods are compared under different training conditions which allow single or multiple pronunciations in the training set, and their performance is evaluated in terms of recall and precision measures.

Key words: P2P conversion, pronunciation lexicon, SMT, Moses, pivot paraphrasing

1 Introduction

Pronunciation variation is one of the factors that influences the performance of an automatic speech recognition (ASR) system, especially for spontaneous speech. Predicting pronunciation variations, that is, alternative pronunciations observed for a linguistically identical word, is a complicated problem and depends on a number of factors, such as the linguistic origin of the speaker, the speaker's education and socio-economic level, the speaking style and conversational context and the relationship between interlocutors. The construction of a good pronunciation dictionary is thus critical to ensure acceptable ASR performance [8]. Moreover, the number of pronunciation variants that need to be included in a dictionary depends on the system configuration [1].

A variety of methods have been proposed in order to obtain pronunciation variants from the canonical pronunciations (baseforms) of words and can be broadly grouped into data-based and knowledge-based methods. Knowledge-based methods using phonological rules [3], [17], require specific linguistic skills, are not language-independent and do not always capture the irregularities in natural languages. By contrast, the data-driven approaches are based on the idea that given enough examples it should be possible to predict the pronunciation of unseen words (in the grapheme-to-phoneme task) or generate multiple

pronunciations for improved speech recognition. In this paper, the latter task of generation of pronunciation variations is addressed using data-based methods. Other data-based methods proposed in the literature for the modeling of pronunciation variations include the use of neural networks [4], confusion tables [14] and decision trees [16] or automatic generation of rules for phoneme-to-phoneme conversion [6]. All these methods predict a phoneme for each input symbol using the input symbol and its context as features, but ignore any structure in the output. The methods proposed in this paper take advantage of both the input and the output context and can predict variable length phoneme sequences.

The two methods presented in this paper aim to automatically generate pronunciation variants of words for which a canonical pronunciation is available. The first method is based on the simple use of Moses [7], a publicly-available phrase-based statistical machine translation tool, as a phoneme-to-phoneme converter to generate an n-best list of pronunciation variants. The second method is based on a paraphrase method that uses bilingual parallel corpora and is founded 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, sequences of modified phonemes found in pronunciation variants are identified using a sequence of graphemes in the corresponding word as a pivot.

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 in terms of precision and recall. Conclusions and some discussions for future work are reported in Section 5.

2 Phoneme-to-phoneme conversion

This section presents the two proposed methods in detail and compares their strengths and weaknesses, pointing out their utility in different situations. Both methods aim to produce pronunciation variants of the initial (canonical) phonemic transcription. Since the canonical pronunciations are not explicitly indicated in the master lexicon (see Section 3.1), the longest one is taken as the canonical form since the reduced forms often correspond to variants found in conversational speech. The first method generates pronunciation variants using only the phonemic transcriptions of words, while the second method makes use of both the orthographic and phonemic transcriptions and thereby permits to the system to also benefit from the information provided by the orthographic transcription of a word. As we will see later in the results analysis, this last characteristic of the second method is particularly useful under certain training conditions.

2.1 Moses as a phoneme-to-phoneme converter

Moses has already been proposed for the grapheme-to-phoneme (g2p) conversion [5], [9] task. 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. In the case that the pronunciation dictionary has a reasonable coverage of the language of interest, this method can be successfully used for g2p conversion because it has all 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 from a phrase-based model and allows sub-strings of graphemes to generate phonemes. The phoneme sequence information is modeled by a phoneme n-gram language model that corresponds to the target language model in machine translation. More technical details on the Moses components will be given in the Section 3.2. These properties are also desired for a p2p converter, which has, moreover, higher potential for capturing pronunciation variation phenomena in languages like English, where orthography and pronunciation generally have a looser relationship than in other languages. A second direction explored in this work is based on the idea of seeing the use of SMT tools with a monolingual corpora for paraphrase generation [11] as being analogous to generating pronunciation variants. These similar approaches to two distinct problems led us to the idea of trying to use Moses for p2p conversion. In this case, the source language and the target language are aligned phonemic transcriptions. As the source language we define the canonical pronunciation (the longest one[1]) and as target language itself and/or its variants depending on their existence or not in the different versions of the training set as presented in the Section 3.1.

2.2 Pivot paraphrasing approach

This method is based on the one presented in [2]. Paraphrases are alternative ways of conveying the same information. We can easily see the analogy with multiple pronunciations of the same word. The multiple pronunciations are alternative phonemic expressions of the same orthographic information.

In [2], a bilingual parallel corpus is used to show how paraphrases in one language can be identified using a phrase in another language as a pivot. In the problem of automatic generation of pronunciation variants, a corpus of word-pronunciation pairs is used as the analogy of the aligned bilingual corpus, instead of the pronunciation-pronunciation aligned corpus in the previous method. The idea is to define a paraphrase (pronunciation variant) probability that allows paraphrases (pronunciation variant sequences) extracted from a bilingual parallel corpus to be ranked using translation probabilities, and then rerank the generated pronunciation variants taking the contextual information into account. The translation table that is used is extracted by Moses. In [2], the authors look at the English translation of foreign language phrases, find all occurrences of those foreign phrases, and then look back to determine to what other English phrases they correspond. The other English phrases are seen as potential paraphrases. In the pronunciation generation case, we look at all the entries in the translation table, find the sequences of graphemes to which a sequence of phonemes is trans-

[1] Most of the variants reflect reduced pronunciations found in casual speech.

lated, and then look back to what other sequences of phonemes the particular sequence of graphemes is translated.

Phrase alignments in a parallel corpus are used as pivots between English (pronunciation) paraphrases. These two-way alignments are found using recent phrase-based approaches to statistical machine translation. In the following definitions, f is a graphemic sequence and e_1 and e_2 are phonemic sequences. The paraphrase probability $p(e_2 \mid e_1)$ is assigned in terms of the translation model probabilities $p(f \mid e_1)$ and $p(e_2 \mid f)$. Since $e1$ can be translated as multiple foreign language phrases (graphemic sequences), we sum over f:

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

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

This returns the single best paraphrase, $\hat{e}_2$, irrespective of the context in which e_1 appears. Since, the best paraphrase may depend on information about the sentence that e_1 appears in, the paraphrase probability can be extended to include the sentence S:

$$\hat{e}_2 = \arg\max_{e_2 \neq e_1} p(e_2 \mid e_1, S) \tag{3}$$

This allows the candidate paraphrases to be ranked based on additional contextual information in the sentence S. A simple language model probability is included, which can additionally rank e_2 based on the probability of the sentence formed by substituting e_2 for e_1 in S. The language model is trained on the correct pronunciations of the training set. For the reranking based on the language model, we use the SRI toolkit [13]. Finally, some more pruning is done on the reranked list keeping the maximum of ten, five or one pronunciation variants per canonical pronunciation without changing the order of the elements of the reranked list.

An example of a paraphrase pattern in the pronunciation dictionary is[2]:

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

The alternative pronunciations differ only in the part that can be realized as either **nt** or **n**, while the rest remains the same.

3 Experiments

3.1 Corpus

The LIMSI Master American English dictionary serves as basis of this work. It is a pronunciation dictionary with 187975 word entries (excluding words starting

[2] The phone set used is given in Table 1 of [8].

with numbers) with on average 1.2 pronunciations per word. The pronunciations are represented using a set of 45 phones [8], each phone corresponding to a single character. The dictionary has been created with extensive manual supervision. Each dictionary entry has the orthographic transcription of a word and its pronunciations (one or more). 18% of the words are associated with multiple pronunciations. The majority of words have only one pronunciation, leaving it to the acoustic model to represent the observed variants in the training set that are due to allophonic differences. Moreover, since the dictionary is mostly manually constructed, it is 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 automatically generated variants are likely to be correct (or plausible) even if they are not in the current version of the Master dictionary.

Case distinction is eliminated since in general it does not influence the word's pronunciation, the main exceptions being the few acronyms which have a spoken and spelled form. 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. It should be noted that these last categories are difficult cases for g2p or p2p converters and particular effort has been made to pronounce proper names in text-to-speech synthesis technology [12].

The corpus is randomly split into a training, a development (dev) and a test set. The dev set is necessary for the optimisation of the weights of Moses model as will be later explained (tuning) and the test set is used for the evaluation of the system. This division is based on dictionary entries so that all the pronunciations of a given word will be in the same set. If not, we would have the paradox of training the system with certain pronunciations and asking it to generate only the different pronunciations of the same word found in the test set.

The dev set has 9000 entries and the test set 16000 entries. The original dictionary entries of training, dev and test sets were transformed to have one graphemic transcription-pronunciation pair per entry as opposed to one entry corresponding to the graphemic transcription of a word with all its pronunciation variants. This is to have a format that resembles the aligned parallel texts used for training machine translation models. After transformation, the dev and test sets have 11196 and 19782 distinct entries. All the results are calculated for the same test set, so that their comparison is legitimate. Three different training conditions are compared for the two p2p systems:

1. Train on the entire dictionary. Words may have one or more pronunciations (tr_set).
2. Train only on words with two or more pronunciation variants. All words have multiple pronunciations (tr_set_m).
3. Train on the entire dictionary using only the longest (canonical) pronunciation to have one pronunciation per word (tr_set_l).

At this point, a further preparation of the training set for each method is required. For the method where Moses is used as a p2p converter, a "monolingual" parallel corpus is needed, meaning that both the source language and the target language will have phonemes as elements. The source language is always formed by the canonical pronunciation segmented into phonemes. The target language is formed of the corresponding pronunciations depending on the training condition. For the pivot method, the training set is used as a parallel corpus with one graphemic transcription-pronunciation pair per line with spaces separating characters, in order to use Moses (as in a g2p task) to generate a translation table that will be used to extract paraphrased sequences. Each word is a source sentence with each grapheme being an element of the source sentence and each pronunciation is a target sentence with each phoneme forming an element of the target sentence.

Table 1 gives an overview of the data sets used with the number of entries (distinct pairs) and the average number of pronunciations/word in the three training conditions after preprocessing.

Table 1. *Training conditions*

Training set	Number of entries	Average number prons/word
tr_set	201423	1.2
tr_set_m	67769	2.3
tr_set_l	162974	1.0

It can be seen in Table 1 that there are large differences for the three training conditions. For tr_set_m the number of entries diminishes to one third of the original dictionary. However, the number of pronunciations per word almost doubles. In this case, the extra information given by the canonical pronunciations of words with only one pronunciation is lost, but we allow the systems to change the frequency relationship between the phrases of the canonical pronunciations and the phrases found in pronunciation variants, and see how this influences the generation of pronunciation variants which is our main interest in this work. In the third training condition, only the canonical pronunciation of each word is kept in the training data. This allows us to see if pronunciation variants can be generated even under limited training conditions. For example, this condition corresponds to generating variants from the output of a rule-based g2p system which, if originally developed for speech synthesis, may not model pronunciation variants or to enriching a dictionary with limited pronunciation variants.

3.2 System

The system that is used to train the models in both methods is based on Moses. In the first proposed method, Moses is used as a p2p converter in an one-stage

procedure. Besides the phrase (translation) table, a phoneme-based 5-gram language model is built on the pronunciations in the training set using the SRI toolkit [13]. Moses also calculates a distortion model, but our dictionary does not include a sufficient number of metathesis cases, so the monotonic decoding does not change the final results. Finally, the combination of all components is fully optimized with a minimum error training step (tuning) on the dev set. The tuning strategy we used was the standard Moses training framework based on the maximization of the BLEU score [10]. The optimized weights generated by tuning are added to the configuration file. Moses can also provide an n-best translation list. This list gives the n-best translations of a source string with the distortion, the translation and the language model weights, as well as an overall score for each translation. As stated earlier we keep only the 1-, 5- or 10-best translations (i.e. pronunciation variants) per canonical pronunciation. Some pronunciations have fewer possible variants, in which case all variants are taken.

In the pivot method, generating pronunciation variants is a four-stage procedure. Moses is used in the first stage for g2p conversion and extraction of the translation table. In the second stage, the paraphrased pairs with their probabilities are extracted from the canonical pronunciations of the test set as previously described. The 10-best paraphrases for each input phonemic sequence are extracted with a maximum length of 3 for the extracted paraphrases. In the third stage, the paraphrases are substituted in the canonical pronunciations of the test set for all their occurrences with all the possible combinations (only in the first occurrence, only in the second occurrence, in the first and in the second occurrence, etc.), limiting to 3 the maximum number of occurrences of the same paraphrase in a canonical pronunciation. In the fourth and final stage the generated list of pronunciation variants is reranked based on the context. The context is expressed by the same phoneme-based 5-gram language model used in the first method. The SRI toolkit is used to rerank the multiple pronunciation n-best list modifying its probabilities. As for the first method, the 1-, 5- or 10-best variants are kept for each canonical pronunciation in the ordered list.

4 Evaluation

Different measures have been proposed to evaluate the predictions of pronunciation variants derived from the original "canonical" form. The most frequently used are precision and recall, first introduced in information retrieval [15]. The canonical pronunciations x_i of the test set can have one or more variants y_i (y_i is a set). Moreover, our systems can generate one or more variants $f(x_i)$ ($f(x_i)$ is also a set). Thus, the recall that corresponds to a couple $(y_i, f(x_i))$ is the number of correct generated variants for a canonical pronunciation in the test set divided by the number of correct variants given in the test set for this canonical pronunciation:

$$r_i = \frac{|f(x_i) \cap y_i|}{|y_i|} \tag{4}$$

The precision is the number of correct generated variants divided by the number of generated variants:

$$p_i = \frac{|f(x_i) \cap y_i|}{|f(x_i)|} \tag{5}$$

The total recall is the mean value of the recall of each example:

$$r = \frac{1}{n} \sum_{i=1}^{n} r_i \tag{6}$$

Analogously, the total precision is the mean value of the precision of each example. We refer to the previous definitions as micro-recall and micro-precision respectively. If the examples are normalized by the number of expected variants (correct variants in the reference), the total recall becomes:

$$r = \frac{\sum_{i=1}^{n} |r_i||y_i|}{\sum_{i=1}^{n} |y_i|} \tag{7}$$

In this last case, the macro-recall is defined. Macro-precision is defined analogously. The macro-measures give more weight to the examples with multiple variants, while the micro-measures consider all the examples equally weighted.

It is important to do the evaluation on a pair level and not just consider the error rate of generated pronunciations, to avoid counting as correct a generated pronunciation that does not correspond to the canonical pronunciation it is associated with in the reference. There is always the possibility that our system will generate a pronunciation out of a baseform (i.e. canonical pronunciation) that is not a variant of this baseform, but, however, is a correct variant of another baseform. This is counted as a false generation. Another thing that should be noted is that we prune the canonical pronunciation-pronunciation variant pairs that do not include a new variant. This is important in order to improve the precision because while the pivot method generates only pronunciation variants, when Moses is used as p2p converter it often outputs the canonical pronunciation that was used as input because it learns from the training data that the most probable pronunciation corresponding to a given canonical pronunciation is usually itself. This depends a lot on the training conditions and makes this method inappropriate in certain cases.

To control the precision of our systems, an upper limit is put to the number of n-best variants that are kept in the hypotheses. The 1-, 5- and 10-best variants per canonical pronunciation are generated consecutively. The n-best list is limited to 10 because preliminary studies showed that larger n only slightly improves recall while severely degrades precision. There is quite a bit of overgeneration, since in the 19k pronunciation-pronunciation pairs of the test set there are only 4k pairs with pronunciation variants. This could not be avoided with a random selection of the test set from the original dictionary where only 18% of words have variants as already stated. However, there is the possibility that some of the generated variants which are not in the reference (and therefore counted as errors) could be considered acceptable by a human judge. Evaluating

Table 2. *Results using Moses as phoneme-to-phoneme converter for the 3 training conditions*

Training set	Measure	1-best	5-best	10-best
tr_set	Micro-recall	0.20	0.75	0.83
	Macro-recall	0.19	0.73	0.81
tr_set_m	Micro-recall	0.21	0.75	0.80
	Macro-recall	0.19	0.74	0.80
tr_set_l	Micro-recall	–	0	0
	Macro-recall	–	0	0

Table 3. *Results using the pivot paraphrasing method for the 3 training conditions*

Training set	Measure	1-best	5-best	10-best
tr_set	Micro-recall	0.29	0.60	0.70
	Macro-recall	0.26	0.56	0.66
tr_set_m	Micro-recall	0.25	0.56	0.70
	Macro-recall	0.22	0.53	0.66
tr_set_l	Micro-recall	0.09	0.26	0.38
	Macro-recall	0.09	0.24	0.35

the system by an automatic measure cannot take into account the potential lack of coverage of the reference dictionary.

The two systems, Moses as phoneme-to-phoneme converter (m_p2p) and the pivot paraphrasing method (p_p2p) were tested for the 3 training conditions presented earlier. The results using the two proposed evaluation metrics are shown in Tables 2 and 3 respectively. We only present recall measures in the tables because this is what is of most interest in the particular task. It is 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. However, the best value of precision is often further limited depending upon the number of elements of the n-best list and the overgeneration that cannot be avoided.

As can be expected, for both methods the number of correctly generated variants increases with the size of the n-best list. This is normal not only because the number of hypothesis increases with the size of the n-best list, but also because there are canonical pronunciations in the test set that have more than one variant (approximately 1/6 of the part of the test set with multiple pronunciations) which cannot be captured when an insufficient number of pronunciations is generated.

It is also interesting to compare the results of the first and the second training conditions (whole dictionary vs. keeping only entries with multiple variants) for the two methods. In the second case, the amount of training data is only one third of the original training set. However, the results are more or less the same for

both training conditions. This may be because the information that the model is using to learn how to generate variants is mostly captured by the multiple pronunciations in the training set and less by the fewer variations observed in the canonical pronunciations of one-pronunciation words. What the model is learning in this case is focused on the relationship between the canonical pronunciation and other variants, and therefore has effectively more relevant information and it does not get watered down by the self-production. This may compensate for the reduced amount of data.

A comparative analysis of the two methods can also be made. In the first (whole training set) and in the second (only entries with variants) training conditions, using Moses as a p2p converter gives better results in terms of the generation of pronunciation variants for both micro and macro measures when the 5-best and the 10-best variants are kept. However, when only the 1-best generated pronunciation is kept, the pivot method gives better results. This is due to the generation of canonical pronunciations by Moses when used as p2p converter, which are subsequently removed from the results because they already exist in the input. The number of variants generated by Moses-p2p when only the 1-best is kept are quite limited. This is why, while the recall is lower than that of the pivot method, the precision is higher.

It can be seen that the results change when the training set is limited to the canonical pronunciations only (the third condition). In this case the pivot method manages to produce some results, while for Moses the model fails to generate any variants (this is why the corresponding columns are left empty in Table 2) and all the variants that are in the 5-best or the 10-best lists are false. These results warrant a bit more discussion. The results are promising for the pivot method, because even when the training dictionary has few or no pronunciation variants, the pivot method can still be used to generate some alternative pronunciations. This can be explained by the fact that the pivot method uses also the graphemic information. Even if no variants are included in the training set, it can still find graphemic sequences of words that correspond to different phonemic sequences and consider these phonemic sequences as possible modifications of pronunciations. For example, in the training set the word "autor**out**e" is pronounced "ctor**ut**" and the word "sh**out**ing" is pronounced "s**Wt**IG". These words have the graphemic sequence "**out**" in common which can be used as a pivot between the phonemic sequences "**ut**" and "**Wt**". These phonemic sequences become a paraphrased pair that generates correctly the variants "r**Wt**s" and "r**ut**s" of the word "r**out**es" found in the test set.

This illustrates the difficulty of generating pronunciations in English, because the correspondence between orthographic forms and canonical pronunciations does not follow strict rules which would prevent the pivot method from finding modified phonemic sequences corresponding to the same graphemic sequence. This is not the case when Moses is used as phoneme-to-phoneme converter. When no variants are given to the system, it does not have any additional information in order to be trained for the task of generating multiple pronunciations. It is like trying to train an SMT system without a target language. It can just learn

to align the phonemic sequences with themselves, which is fine for the g2p task, but is not applicable to the generation of variants. In this case, is it wrong to use Moses for this task as it is obvious that it has nothing to learn from the training data.

5 Conclusions

This paper has reported on applying two data-based approaches inspired from research in statistical machine translation to the problem of generating pronunciation variants. One of the objectives of this work was to compare these two approaches to modeling pronunciation variations. The approaches differ in the way that information about pronunciation variation is obtained. The approach using Moses as phoneme-to-phoneme converter takes into account only the information provided by the phonemic transcriptions. The pivot method uses information from both the phonemic and the orthographic transcriptions.

When the full dictionary (that contains words with one or more pronunciations) is used for training, the Moses-based method gives better results than the pivot-based one. This is also the case when training is carried out on only entries with multiple pronunciations. However, when the training dictionary does not contain any pronunciation variants, the Moses-based method cannot be used, while pivot can still learn to generate variants. This is an advantage of the pivot method, and could be useful for languages without well-developed multiple-pronunciation dictionaries. This arises from the use of information provided by the orthographic transcription by the pivot method. An interesting follow-up study is to use the pivot method to propose variants as a post-processing step to a g2p system. Another case to study is the influence of the p2p converter on the results of a g2p converter if their output n-best lists are combined. We have started some preliminary experiments in these directions with promising results.

In future work we will evaluate the proposed methods at generating pronunciations and variants for proper names, which are the most difficult cases to handle. These also account for the majority of words that need to be added to a dictionary once a reasonably sized one is available for the given language.

Another important outstanding issue concerns the proper way to evaluate the ability of a system to generate pronunciation variants. In this work, recall and precision have been used, however other measures such as phoneme accuracy can also be applied. In this case it may be appropriate to have phone-class dependent penalties with certain confusions being more important than others. In order to improve the precision, the n-best lists need to be more heavily pruned. One direction to explore is using audio data to remove pronunciations, however this can only apply to words found in the audio data.

The ultimate test of course is how the variants affect the accuracy of a speech-to-text transcription system.

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. Adda-Decker, M. and Lamel, L.: Pronunciation variants across system configuration, language and speaking style. In: Speech Communication, vol. 29, pp. 8398 (1999)
2. Bannard, C., Callison-Burch, C.: Paraphrasing with bilingual parallel corpora. In: Proc. of ACL, pp. 597 - 604 (2005)
3. Divay, M., Vitale, A.-J.: Algorithms for grapheme-phoneme translation for English and French: Applications for database searches and speech synthesis. In: Computational linguistics, vol. 23, n^o4, pp. 495-523 (1997)
4. Fukada, T., Yoshimura, T., Sagisaka, Y.: Automatic generation of multiple pronunciations based on neural networks. In: Speech communication, vol.27,n^o1,pp.63-73 (1999)
5. Gerosa, M., Federico, M.: Coping with out-of-vocabulary words:open versus huge vocabulary ASR. In: ICASSP, pp. 4313-4316 (2009)
6. Heuvel, H. van de, Reveil, B., Martens, J.-P.: Pronunciation-based ASR for names. In: Proc of Interspeech, pp. 2991-2994 (2009)
7. Koehn, P. et al.: Moses: Open source toolkit for statistical machine translation. In: Proc. of ACL (2007)
8. Lamel, L. and Adda, G.: On designing pronunciation lexicons for large vocabulary, continuous speech recognition. In: Proc. ICSLP-96, pp. 6-9 (1996)
9. Laurent, A., Deleglise, P., Meignier, S.: Grapheme to phoneme conversion using a SMT system. In Proc. of Interspeech (2009)
10. Papineni, K., Roukos, S., Ward, T. and Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proc. of ACL, pp. 311-318 (2002)
11. Quirk, C., Brockett, C., Dolan, W.: Monolingual Machine Translation for Paraphrase Generation. In: Proc. of EMNLP, pp. 142-9 (2004)
12. Spiegel, M. F.: Using the ORATOR synthesizer for a public reverse-directory service:design, lessons, and recommendations. In: EUROSPEECH'93, pp. 1897-1900 (1993)
13. Stolcke, A.: SRILM-An extensible language modeling toolkit. In Proc. ICSLP-02, vol. 2, pp. 901-904 (2002)
14. Tsai, M.-Y., Chou, F.-C., and Lee L.-S.: Pronunciation modeling with reduced confusion for mandarin chinese using a three-stage framework. In: IEEE Transactions on audio, speech and language processing, vol.15, n^o2, pp. 661-675 (2007)
15. Van Rijsbergen, C.J.: Information Retrieval, Butterworths, London, UK. (1979)
16. Weintraub, M., Fosler, E., Galles,C., Kao,Y.-H., Khudanpur, S., Saraclar, M. and Wegmann, S.: WS96 project report:Automatic learning of word pronunciation from data. In: JHU Workshop Pronunciation Group (1996)
17. Wester, M.: Pronunciation modeling for ASR- Knowledge-based and data-driven methods. In: Comput. Speech Lang., pp. 69-85 (2003)