INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Developing Pronunciation Models in New Languages Faster by Exploiting Common Grapheme-to-Phoneme Correspondences Across Languages

Harry Bleyan[†], Sandy Ritchie[†], Jonas Fromseier Mortensen, Daan van Esch

Google

{askharry, sandyritchie, jfmortensen, dvanesch}@google.com

Abstract

We discuss two methods that let us easily create grapheme-to-phoneme (G2P) conversion systems for languages without any human-curated pronunciation lexicons, as long as we know the phoneme inventory of the target language and as long as we have some pronunciation lexicons for other languages written in the same script. We use these resources to infer what grapheme-to-phoneme correspondences we would expect, and predict pronunciations for words in the target language with minimal or no language-specific human work. Our first approach uses finite-state transducers, while our second approach uses a sequence-to-sequence neural network. Our G2P models reach high degrees of accuracy, and can be used for various applications, e.g. in developing an automatic speech recognition system. Our methods greatly simplify a task that has historically required extensive manual labor.

Index Terms: pronunciation models, speech processing, machine learning

1. Introduction

Over 7,000 languages are spoken in our world today, out of which nearly 4,000 are known to have a written form [1]. And in fact, text data can easily be found online in well over 2,000 languages [2, 3]. However, automatic speech recognition (ASR) systems are available only for around 100 language varieties. Creating an ASR system in the finite-state transduction framework summarized in [4] requires an acoustic model, a grapheme-to-phoneme (G2P) conversion component, and a language model. [3] described a scalable approach to creating language models across many languages. In this paper, we focus on coming up with equally scalable approaches to creating the G2P component, in the vein of the work done by [5, 6].

Specifically, we will describe two approaches we have found to be promising. First, we will describe an automatic system to propose G2P rules for a new language, significantly accelerating the work of linguistic experts who would otherwise be tasked with writing these G2P rules from scratch. This method operates in the finite-state transduction framework our linguistic experts typically use to express G2P rules. Second, we will describe a multilingual machine-learning G2P model that generalizes to unseen languages. Both approaches significantly reduce the amount of time and effort required to create the G2P component for a new language (see also [7]), which makes it easier to build voice technologies such as ASR systems in more languages.

2. Background

G2P conversion is an essential, and typically costly, step in the process of building an automatic speech recognition (ASR) system for a new language. Broadly, there are two common approaches to handling G2P. The least resource-intensive way to build the G2P component for a new language, which works especially well for languages with transparent orthographies (where there is a relatively straightforward relationship between graphemes and phonemes), is to adopt a rule-based approach. In the rule-based approach, linguistic experts research the orthographic system used in the target language (e.g. consulting the charts on Wikipedia or Omniglot), and express the G2P correspondences in machine-readable format, e.g. using ICU Transform rules [8], Thrax [9], or Pynini [10].

A much costlier approach is to hand-curate pronunciation dictionaries, and to combine these dictionaries with machine-learning grapheme-to-phoneme models learned from them, e.g. using Phonetisaurus [11]. While this approach is costlier, manual curation of pronunciation dictionaries can allow for far more accurate transcriptions than rule-based approaches, depending on how transparent the orthography of the target language is. Since more accurate G2P conversion typically yields significant improvements in quality for ASR systems, this investment may well be worthwhile for some languages, but given the linguistic expertise it requires, it is infeasible to curate pronunciation dictionaries manually across hundreds, let alone thousands, of languages.

Even if there are thousands of languages, each with their own orthographic standard and G2P rules, the number of commonly used writing systems is drastically smaller. The Latin script (ISO 15924 "Latn") is currently the most widely used script in the world, and is used by at least 1,800 of the world's written languages [12]. Other widely used scripts include the Arabic script ("Arab") and its derivatives; Cyrillic ("Cyrl"); and the Devanagari script ("Deva"), which is a member of the Brahmic family of scripts. Generally, each language has a language-specific orthography that uses a subset of the graphemes available in a given script. Each language also has some set of phonemes (sounds) in its phonological system, and the language-specific orthography can be thought of as a system governing how these sounds are to be rendered into graphemes. The task of a G2P conversion system for a given language is to correctly and automatically assign a phonemic transcription to a sequence of graphemes.

The challenge is that orthographic systems, i.e. the correspondences between graphemes and phonemes, differ from language to language. For example, using IPA notation for phonemic transcriptions, the grapheme sequence <variation> is pronounced as /vɛrieʃən/ in English, but as /varjasjɔ̃/ in French.

However, even if the full set of G2P correspondences is rarely the same between even closely related languages, there are parts which seem to be relatively predictable. For example, a literate speaker of one Germanic language written in the Latin alphabet, such as English, can typically make a reasonable guess of how a word in another Germanic language written

[†] Equal contribution.

in the Latin alphabet would be pronounced. This makes sense, given that many correspondences do not change significantly: for example, it would be very unusual for the grapheme <m> to represent a vowel phoneme such as /æ/. That is not to say it is impossible, but just that it would be a rather unusual correspondence; it is exactly these tendencies that we exploit in our work here.

In another similar example, it would also be unusual for the grapheme <r> to be used for a sound that isn't one of the R-like rhotic consonants (except, perhaps, rhotic schwa). Now, imagine we were presented with a word in some target language that contains an <r>. Under the assumption that the corresponding phoneme would have to be one of the rhotic consonants, the challenge would be determining which of the R-like consonants to use in generating a phonemic transcription for this target language. Fortunately, we have access to databases such as Phoible [13], containing machine-readable phoneme inventories for more than 1,600 languages, which are easily accessible using the FonBund library [14]. In this example, these phoneme inventories can help us constrain our search for the correct rhotic consonant in this target language.

If we can design a system that can exploit these cross-language G2P regularities, combining knowledge in databases such as Phoible with our existing knowledge of G2P correspondences in some languages that use the same script as our target language, we should be able to create G2P systems for new languages in a relatively straightforward way. We assume, here, that we have access to G2P systems for some number of languages written in the same script, which is frequently going to be the case given the relatively limited number of scripts in common use today. (For languages written in less commonly used scripts, we can still use the traditional approaches described earlier; one of the benefits of accelerating G2P development for the vast majority of languages is that our linguistic team can focus on correctly handling less commonly used scripts, where human input is needed more.)

3. Inducing rule-based G2P finite-state transducers

Perhaps the simplest way to make a G2P system is to create a list of all the graphemes in the target language, e.g. from a source such as [3]. Then, for each grapheme, we assign the phoneme that most commonly corresponds to this grapheme in other languages for which we do have human-curated pronunciations or rule-based G2P, restricting ourselves to the phonemes that are known to appear in the target language. For example, the grapheme <b> is used to represent the phoneme /b/ across many languages. If the target language has the grapheme <b> and the phoneme /b/, we can reasonably assume that this grapheme will map to this phoneme in the target language. This approach has some limitations, notably that it assumes that the target-language orthography uses a 1:1 mapping to one specific phoneme for each grapheme, but as we will see, this approach does get us quite far.

Concretely, the basic requirements for our induction system are a list of graphemes and a list of phonemes for the target language, as well as some pre-existing finite state transducers (FSTs) which encode G2P mappings in other languages. These G2P FSTs can be trained using a framework such as Phonetisaurus [11], on data sets such as [15], or they can be derived from running a wordlist through [5] and then creating an FST. In our experiments, we used internal lexicons and a trainer similar to Phonetisaurus. Our induction system can also accept graphone specifications [16], which we turn into FSTs as well. The lists of graphemes and phonemes serve as the targets that we want to map between, and the pre-existing FSTs are the data from which we can induce likely G2P mappings in the target.

The grapheme and phoneme lists are converted to finite state transducers using the string map function in Pynini [10]. This converts the lists to acceptors. These FSTs are then composed with a pre-existing G2P FST in the following configuration:

$$\text{graphemes}_{\text{L1}} \circ \text{g2p}_{\text{L}n} \circ \text{phonemes}_{\text{L1}}$$

Composing the grapheme and phoneme FSTs with a pre-existing G2P FST allows us to predict the G2P mappings in the target, based on the mappings in the source FST. Repeating this process with a large number of pre-existing G2P FSTs increases the chances of inducing mappings for rare or low-frequency graphemes and phonemes. It also allows us to take a 'majority vote' in cases where a grapheme maps to a number of different phonemes in different languages. Once the target graphemes and phonemes have been composed with all pre-existing G2P FSTs, we use a counting tool to retain only the most common G2P mapping for each grapheme. For example, the <c> grapheme in Latin script maps variously to /k/, /tʃ/ or other phonemes depending on the language. If we have seen four mappings from <c> to /k/, but only two from <c> to /tʃ/ in all the pre-existing FSTs, then the most common rewrite is selected and all others are removed.

This approach does not predict contextual correspondences between graphemes and phonemes. For example, a <c> grapheme might correspond to the phoneme /k/ at the beginning of a word, but to /tʃ/ in medial and final position, either in the target language, or source language, or both. Since we are inducing one-to-one correspondences, this kind of contextually-aware mapping is not possible using this approach, though it could be extended to also use graphone specifications which can map sequences of graphemes to sequences of phonemes. This would also allow us to map multigraphs (i.e. digraphs like <sh>) to monophones like /ʃ/, and monographs like <o> to di- or triphones like /oʊ/.

After the induction process has been carried out, there are usually some unmapped graphemes and phonemes left over. These are retained as separate lists. The induced G2P mappings are also divided into consonant and vowel sets by comparing the phonemes with pre-existing lists of consonant and vowel phonemes. These four sets: unmapped graphemes, unmapped phonemes, consonant mappings and vowel mappings, are then written to a Thrax grammar file and combined with other rules and methods like insertion of stress and syllabification markers to produce phonemic transcriptions of written words. The resulting grammar then only needs light editing by a linguist to be compiled as an FST which can be used to generate G2P pronunciations for written words in the target language.

To test the accuracy of the G2P induction method, we compared the G2P mappings and pronunciations generated by the system with hand-written pronunciation rules in 24 languages. In each case, the tool was run on a language for which we already have human-curated pronunciation rules, and the quantity and quality of the G2P mappings were scored. The results are set out in Table 1. The grapheme counts only include lower-case variants of graphemes, and grammatical and/or adjoining punctuation such as hyphens and apostrophes are not included.

Table 1: *Accuracy of G2P induction*

Script	**Number of languages**	**(Average) number of graphemes**	**% mapped to correct phoneme**
Arabic	3	58	68
Cyrillic	1	42	83
Devanagari	3	57	58
Latin	17	34	86

The results in Table 1 indicate that despite the limitations outlined above, the script can predict G2P mappings in an unknown locale with a fairly high degree of accuracy. This greatly eases the cognitive load on the linguist working on the pronunciation rules grammar, allowing them to spend more time on more complex aspects of G2P mapping.

Future developments for the system will include better support for contextually-aware mappings. This could be achieved by gathering small datasets from e.g. the ASJP database [17], which lists pronunciations of words in the Swadesh wordlist [18] in thousands of languages, or from resources that provide phonotactic constraints such as the World Phonotactics Database [19]; the StressTyp database [20]; the UCLA Phonological Segment Inventory Database [21]; and the Lyon-Albuquerque Phonological Systems Databases [22].

4. Multilingual G2P models which generalize to unseen languages

Inducing rule-based FST systems as described above yields significant productivity gains for the linguists who are tasked with writing rule-based G2P grammars for a new language. However, human input is still required, and some assumptions may be less than ideal: for example, it is hard to deal with multiple graphemes corresponding to a single phoneme. For example, the Catalan word `plorarono` happens to have a one-to-one mapping to the pronunciation /plorarono/, but the French word `permettrais` has a more complex mapping /permetre/, which our induction method cannot easily suggest. We wondered if a recurrent neural network (RNN) G2P model might be able to do a better job, as in [23, 24]. In the RNN G2P approach, a set of G2P rules for the language is inferred from a human-curated pronunciation lexicon. The ability of RNNs to generalize a variety of patterns in the data means that, given previously unseen words in a language, an RNN-based system can do a better job at phonemic transcription [23] than traditional G2P machine-learning FSTs with more rigid alignment rules, such as [25].

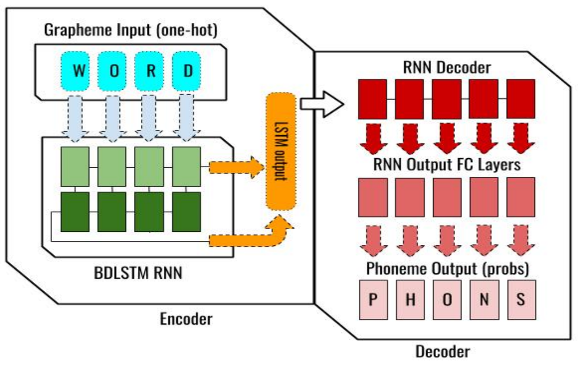

Figure 1: *Diagram of standard end-to-end G2P system.*

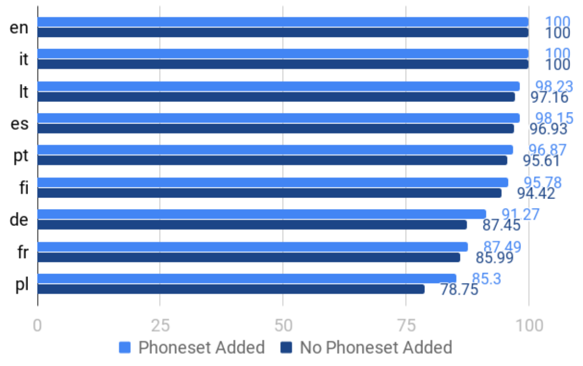

Figure 2: *Percentage of generated phoneme types present in the target language.*

The diagram in Figure 1 shows an example of an RNN G2P system that contains a Bidirectional LSTM (BLSTM) encoder and a Multilayered RNN (MRNN) decoder, as used in [23]. In short, the one-hot representation of input graphemes is padded, converted into a matrix of grapheme embeddings, and passed into a BLSTM, which creates an internal representation of the input word. This vector is then passed to the decoder MRNN, which unrolls the internal representation vector and predicts the output phoneme at every unroll step, terminating after predicting the end-of-word label.

4.1. Phoneset-enabled RNN G2P for generalization

RNN-based G2P models generally perform well on languages from which they were trained [23]. For example, the implementation described in Figure 1 gets up to 87% word-level accuracy on an Italian transcription task after being trained on a large Italian phonetic dictionary consisting of 900K words and their phonemic transcriptions. However, for the purposes of building G2P models for entirely unseen languages, existing training methods do not suffice. While current RNN models trained on single languages perform well on inference tasks within those specific languages, they do not generalize well to the task of generating phonemic transcriptions for orthographic forms in previously unseen languages.

A naive approach is to concatenate all the training data from different languages and train on that cumulative set in order to cover a wider variety of possible G2P mappings in the training data, but the resulting phonemic transcriptions do not work well for any one specific target language (as seen in the right-hand side of Figure 4). We trained such a model on 30 languages in the Latin script, and evaluated on the held-out test languages. As expected, the resulting phonemic transcriptions frequently contain phonemes that do not appear in the target language, as reflected in Figure 2.

To address this issue, we made phoneme sets of the languages available to the model at training time. Using open data sets like Phoible [13], we can aggregate and pass in the phoneme set of the language both at training and inference time.

The new model closely resembles the traditional RNN architecture, with the main difference being the special phoneset vector. This vector is made up of a set of phoneme embeddings assembled from the phonemes present in the phoneset of the target language. The embeddings of the phonemes are trained along with the model. Our hope is that the phoneset space will contribute additional information in the form of "boundaries" between the learned language-specific G2P rules within the en-

Table 2: *Composition of input data*

Column in Data	Example Content
Graphemes	zwischenziel
Pronunciation	ts v ɪ ʃ ə n ts iː l
Phoneset	ɒ ɵ ɘ _ _ ə _ _ _ _ ç ð ɛ _ _ _ ɪ _ _ _ _ _ _ ŋ ɔ _ ɔɪ _ _ ɾ ʃ _ θ ʊ _ _ _ ɤ ʒ a aː aɪ aʊ _ _ b _ _ _ d _ _ _ dʒ _ _ _ _ e _ _ _ _ f _ g h _ _ iː _ _ j _ _ k _ _ l _ _ m _ n _ _ o _ _ _ p _ pf _ _ _ s _ _ _ _ t _ _ _ tʃ _ _ _ _ ts u _ v _ _ w _ x _ y _ z _ _ _ _ _ _

coder's representation of the input word. These embeddings are concatenated end-to-end along with the grapheme embeddings and passed into the encoder at training and inference time. Inference for an unseen language merely requires the availability of the phoneme inventory for the target language; it is also a requirement that all the graphemes in the unseen language must have been observed at least once in one of the training languages.

4.2. Experimental setup across languages

The training and test data sets for each language were derived from our internal human-curated pronunciation lexicons. For most languages, the lexicon has loan and foreign words, including words written in foreign alphabets. Processing and evaluation do not distinguish native and foreign words. For the results listed below, the lexicons were shuffled, and every tenth item was reserved for testing. For multilingual training, lexicons from every trained language were combined before shuffling and test data allocation.

Table 2 shows the 3 columns of a single line from the data passed into the model at training time. The columns are tab-separated and consist of the grapheme representation of the word, its space-separated phonetic transcription, and a padded space-separated list of phonemes present in the phoneset of the language the word is in (German in this case), where the _ character serves as the padding symbol for global phonemes not present in the target language.

At test time, only the graphemes and phoneset are put into the model. If a grapheme in the test language does not appear in the training set, we attempt to strip diacritics from it using the Unicode library. If this does not result in a previously seen grapheme, we avoid generating a pronunciation for that word.

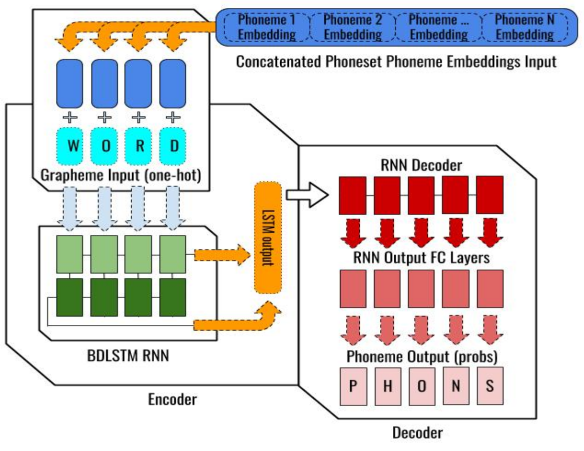

Figure 3: *Phoneset-enabled end-to-end G2P system.*

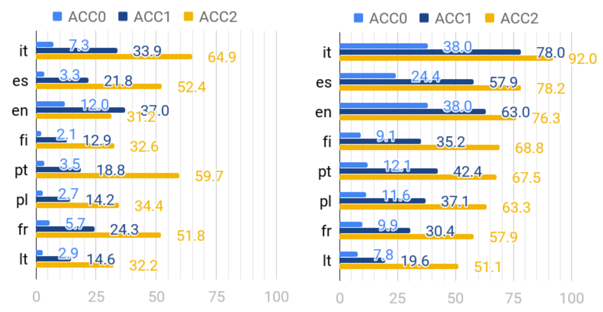

Figure 4: *Performance of traditional (left) vs Phoneset-enabled (right) end-to-end multilingual G2P models on unseen languages. ACC0, ACC1 and ACC2 metrics represent the accuracy of predictions with 0, 1 and 2 phoneme errors per word, respectively.*

4.3. Phoneset-enabled LSTM experiments

Upon running the experiments on a set of data consisting of 30 Latin-script languages, we were able to confirm that the addition of a phoneset significantly improves the performance of the models on unseen languages.

Figure 4 shows the performance of a model trained on the 30 aforementioned languages when tested against 8 languages from differing language groups. For each of the 8 models we hold out the test language from the training set, and then evaluate on the held-out language. On the right are the results of phoneset-enabled models, on the left of those without phoneset additions.

While the accuracy rates on unseen languages do not yield performance close to a language-specific model trained specifically on the held-out language, we observe that relatively low phoneme error rates can be obtained. Confirming our hypothesis, accuracy improves significantly when the model is trained including phonesets and provided with the phoneset for the unseen language at inference time. We believe that future work along these lines could yield improved systems that generalize well to unseen languages, either through exploring the hyper-parameter space more thoroughly, or by coming up with better ways to encode the phoneme-inventory data into the models and the decoder. We could also constrain the decoder to emit only valid phonemes in the target language.

5. Conclusion

In this paper we described two methods that enable us to build relatively accurate G2P conversion systems for new languages. While neither the rule-based nor the phoneset-enabled end-to-end G2P models will match a human-curated lexicon in terms of quality, we are able to obtain a fairly high level of accuracy at much lower cost than traditional approaches such as a fully hand-written set of G2P rules, or a human-curated lexicon. Our approaches can be applied when as the target language shares a writing system with a number of other languages for which pronunciation lexicons are available, and as long as the phoneme inventory of the target language is known. Given that these conditions are met in many of the world's languages, the methods we described could be helpful in bringing language technologies such as speech recognition to more languages.

6. References

[1] D. M. Eberhard, G. F. Simons, and C. D. Fennig, Eds., *Ethnologue: Languages of the World*, 22nd ed. Dallas, Texas: SIL International, 2019.

[2] K. P. Scannell, "The Crúbadán Project: Corpus building for under-resourced languages," 2007.

[3] M. Prasad, T. Breiner, and D. van Esch, "Mining training data for language modeling across the world's languages," in *SLTU*, 2018, pp. 61–65.

[4] M. Mohri, F. C. N. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Handbook on Speech Processing and Speech Communication, Part E: Speech recognition*, L. Rabiner and F. Juang, Eds. Heidelberg, Germany: Springer-Verlag, 2008.

[5] D. R. Mortensen, S. Dalmia, and P. Littell, "Epitran: Precision g2p for many languages," in *LREC*, 2018, pp. 2710–2714.

[6] A. Deri and K. Knight, "Grapheme-to-phoneme models for (almost) any language," in *ACL*, 2016, pp. 399–408.

[7] D. Sharma, "On training and evaluation of grapheme-to-phoneme mappings with limited data," in *Interspeech*, 2018, pp. 2858–2862.

[8] Unicode, Inc, "International components for unicode," https://github.com/unicode-org/icu, 2016.

[9] OpenFST Org, "Opengrm thrax grammar development tools," http://www.openfst.org/twiki/bin/view/GRM/Thrax.

[10] K. Gorman, "Pynini: A Python library for weighted finite-state grammar compilation," in *ACL Workshop on Statistical NLP and Weighted Automata*, 2016, pp. 75–80.

[11] J. Novak *et al.*, "Phonetisaurus," http://code.google.com/p/phonetisaurus.

[12] T. Breiner, C. Nguyen, D. van Esch, and J. O'Brien, "Automatic keyboard layout design for low-resource latin-script languages," *CoRR*, vol. abs/1901.06039, 2019. [Online]. Available: http://arxiv.org/abs/1901.06039

[13] S. Moran, D. McCloy, and R. Wright, Eds., *PHOIBLE Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology, 2014. [Online]. Available: https://phoible.org/

[14] A. Gutkin, M. Jansche, and T. Merkulova, "Fonbund: A library for combining cross-lingual phonological segment data," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 7-12 May 2018, Miyazaki, Japan, 2018, pp. 2236–2240. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2018/pdf/8889.pdf

[15] K. Lenzo *et al.*, "The cmu pronouncing dictionary," http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[16] M. Jansche, "Computer-aided quality assurance of an Icelandic pronunciation dictionary," in *LREC*, 2014.

[17] S. Wichmann, E. W. Holman, and C. H. Brown, "The ASJP Database," https://asjp.clld.org/, 2018.

[18] M. Swadesh, "Lexicostatistic dating of prehistoric ethnic contacts," in *Proceedings American Philosophical Society*, vol. 96, 1952.

[19] M. Donohue, R. Hetherington, J. McElvenny, and V. Dawson, "World phonotactics database," http://phonotactics.anu.edu.au, 2013.

[20] R. Goedemans, J. Heinz, and H. van der Hulst, "StressTyp2," http://st2.ullet.net/, 2019.

[21] I. Maddieson and K. Precoda, "UCLA Phonological Segment Inventory Database," http://web.phonetik.uni-frankfurt.de/upsid.html, 2014.

[22] I. Maddieson, S. Flavier, E. Marsico, and F. Pellegrino, "Lyon-Albuquerque Phonological Systems Databases," http://www.lapsyd.ddl.cnrs.fr/lapsyd/, 2014.

[23] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[24] K. Yao and G. Zweig, "Sequence-to-sequence neural net models for grapheme-to-phoneme conversion," *CoRR*, vol. abs/1506.00196, 2015. [Online]. Available: http://arxiv.org/abs/1506.00196

[25] M. Bisani and H. Ney, "Joint-Sequence Models for Grapheme-to-Phoneme Conversion," *Speech Communication*, vol. 50, no. 5, p. 434, Mar. 2008. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00499203