A phonetization approach for the forced-alignment task

Brigitte Bigi

Laboratoire Parole et Langage, CNRS, Aix-Marseille Université,
5, avenue Pasteur, BP80975, 13604 Aix-en-Provence, France
brigitte.bigi@lpl-aix.fr

Abstract

The phonetization of text corpora requires a sequence of processing steps and resources in order to convert a normalized text in its constituent phones and then to directly exploit it by a given application. This paper presents a generic approach for text phonetization and concentrates on the aspects of phonetizing unknown words, which serve to develop a phonetizer in the context of forced-alignement application. It is a dictionary-based approach, which is as language-independent as possible: this approach is applied to French, English, Vietnamese, Khmer and Pinyin for Chinese. The tool with linked resources are distributed under the terms of the GPL license.

Keywords: phonetization, graphemes-phonemes, unknown words, LRL

1. Introduction

Phonetic transcription of text is an indispensable component of text-to-speech (TTS) systems and is used in acoustic modeling for automatic speech recognition (ASR) and other natural language processing applications. Phonetic transcription can be implemented in many ways, often roughly classified into dictionary-based and rule-based strategies, although many intermediate solutions exist. The “Forced Alignment”’ task included both phonetization and alignment tasks: phonetization is the process of representing sounds by phonetic signs; alignment is the process of aligning speech with these sounds. The phonetization step takes as input the transcription of the speech signal and produces the supposed pronunciation.

Clearly, there are different ways to pronounce the same utterance. Different speakers have different accents and tend to speak at different rates. When a speech corpus is transcribed into a written text, the transcriber is immediately confronted with the following question: how to reflect the orality of the corpus? Conventions are then designed to provide rules for writing speech corpora. These conventions establish phenomena to transcribe and also how to annotate them. There are commonly two types of Speech Corpora. First is related to “Read Speech” which includes book excerpts, broadcast news, lists of words, sequences of numbers. Second is often named as “Spontaneous Speech” which includes dialogs - between two or more people (includes meetings), narratives - a person telling a story, map-tasks - one person explains a route on a map to another, appointment-tasks - two people try to find a common meeting time based on individual schedules. One of the characteristics of Spontaneous Speech is an important gap between a word’s phonological form and its phonetic realizations. Specific realization due to elision or reduction processes (for example, in Italian, *perchè* pronounced as /b e k/, *il videotelefono* as /jo d e l e f/) are frequent in spontaneous data. It also presents other types of phenomena such as non-standard elisions, substitutions or addition of phonemes which intervene in the automatic phonetization and alignment tasks.

After the state-of-the-art, we describe our phonetization system that implements a language-independent algorithm to phonetize unknown words. We also briefly describe the aligner. Then, we propose evaluations of the phonetization system.

2. State-of-the-art

Grapheme-to-phoneme conversion is a complex task, for which a number of diverse solutions have been proposed. It is a structure prediction task; both the input and output are structured, consisting of sequences of letters and phonemes, respectively. Phonetic transcription of text is an indispensable component of text-to-speech systems and is used in acoustic modeling for speech recognition and other natural language processing applications. Converting from written text into actual sounds, for any language, cause several problems that have their origins in the relative lack of correspondence between the spelling of the lexical items and their sound contents. While Grapheme-to-phoneme conversion has been heavily studied for Text-To-Speech systems, it has been very little for Automatic Speech Recognition and not at all for forced-alignment (we suppose this is because forced-alignment is often considered as an ASR sub-problem).

2.1. Text-To-Speech synthesis

Grapheme-to-Phoneme conversion is necessary for determining the *canonical* phonemic transcription of a word from its orthography in a Text-To-Speech system. It is commonly implemented in the form of a Letter-To-Sound module which is responsible for the automatic determination of the phonetic transcription of the incoming text. In this context, the Letter-To-Sound module can not simply perform the equivalent of a dictionary look-up. As mentioned in (Dutoit, 1997), this is for the following reasons:

1/ Dictionaries in TTS systems only refer to word roots

pronunciation: they do not include morphological variations (i.e. plural, feminine, conjugations).
2/ Languages contain heterophonic homographs, i.e. words that are pronounced differently even though they have the same spelling. The appropriate pronunciation could often be determined by using a Part-of-Speech Tagger. 3/ "Pronunciation dictionaries merely provide something that is closer to a phonemic transcription than from a phonetic one (i.e. they refer to phonemes rather than to phones)."
4/ Words embedded into sentences are not pronounced as if they were isolated.
5/ "Not all words can be found in a phonetic dictionary: the pronunciation of new words and of many proper names has to be deduced from the one of already known words."
The Letter-To-Sound modules can be implemented in many ways, often roughly classified into dictionary-based and rule-based strategies, although many intermediate solutions exist. Dictionary based solutions consist in storing a maximum of phonological knowledge in a lexicon and rule based systems consist on rules that are based on inference approaches or proposed by expert linguists. Both dictionary-based and rule-based methods on Grapheme-to-Phoneme conversion have their own advantages and limitations. Looking a word up in a lexicon is relatively cheap computationally, whereas most algorithms for rule-based systems use considerably more processor resource to produce the phoneme sequence. Furthermore, a large sized phonetic dictionary and complex morphophonemic rules are required for the dictionary-based method and the Letter-To-Sound rule-based method itself cannot model the complete morphophonemic constraints.

Initially, dictionary based approach was developed in the MITTALK system (Allen et al., 1987) where a dictionary of up to 12,000 morphemes covered about 95% of the input words. In the same way, the AT&T Bell Laboratories TTS system followed the same guideline (Levinson et al., 1993), with an augmented morpheme lexicon of 43,000 morphemes.

At its first stage, (Divay and Guyomard, 1977) proposed a transformation rules system for French. The rules system is based on the application of a partially ordered set of phonological rules: left-hand side of each rule indicates the graphemes involved by the rule, right-hand side of each rule specifies the corresponding phonemes and possibly the preceding and succeeding graphemic context. Exceptional pronunciation rules are first examined in the set and the last examined rules are the more general ones. Since the 1990s, considerable efforts have been made towards designing sets of rules with a very wide coverage (starting from computerized dictionaries and adding rules and exceptions until all words are covered, for various languages. Often rule-based Grapheme-to-Phoneme systems also incorporate a dictionary as an exception list. In (Belrhali et al., 1992), a descriptive language permits the integration of rules and lexica into a text-to-phonetics grammar. A minimal grammar, constituting the core of the phonetization process, has been enlarged by systematically exploring a representative lexicon of French. A clearly disadvantageous consequence of such a knowledge-based strategy is that it requires a large amount of handcrafting of linguistic rules (and data). In contrast to the knowledge-based approach outlined above, the data-driven approach to grapheme-to-phoneme conversion is based on the idea that given enough examples it should be possible to predict the pronunciation of unseen words purely by analogy. Such systems are based on a training stage from aligned data, alignments between letters and phonemes can be discovered reliably with unsupervised generative models. Given such an alignment, Letter-To-Sound conversion can be viewed either as a sequence of classification problems, or as a sequence modeling problem. In the classification approach, like in (Daelemans and Van Den Bosch, 1997; Galescu and Allen, 2001), rules are trained from a given set of examples in a language and the Grapheme-to-Phoneme system was automatically produced for that language. To train rules, the training data consists of letter strings paired with phoneme strings, without explicit links connecting individual letter to sound. These systems predict a phoneme for each input letter, using the letter and its context as features. In the sequence modeling approach, various models was proposed. In (Taylor, 2005), a supervised Hidden Markov Model is applied, where phonemes are the hidden states and graphemes the observations. Several other approaches have been adopted, such as Kohonen's concept (Torkkola, 1993) finite state transducers (Caseiro et al., 2002), etc. For a review, see (Bisani and Ney, 2008).

Finally, there are many competing techniques for Letter-To-Sound conversion for TTS systems and the system developer must make a rational selection among them. For comparison and evaluation of different methods, we refer to (Damper et al., 1998), (Yvon et al., 1998) and (Jiampojamarn et al., 2008). In (Damper et al., 1998), authors report a comparative assessment of the competitor methods of Letter-To-Sound rules (for English only), pronunciation by analogy, feedforward neural networks and a k-nearest neighbor method, with respect to their success at automatic phonemization. (Yvon et al., 1998) reports on a cooperative international evaluation of Grapheme-To-Phoneme conversion for Text-To-Speech in French. The systems involved was all relying on a rule-based approach. The evaluation was performed on the phonemization of 12000 sentences. Overall, the eight systems fared relatively well: they all achieve at least 97% phonemes correct. Difficulties are due to proper names, heterophonous homographs, pre-processing, schwa and liaison. Recently, (Jiampojamarn et al., 2008) proposed a discriminative structure-prediction model and compared performances with six publicly available data sets representing four different languages: English, German and

Dutch CELEX, French Brulex, English Nettalk and English CMUDict data sets. The results for the CMUDict range from 57.8% to 71.99% accuracy.

2.2. Automatic Speech Recognition

Grapheme-to-phoneme technology is also useful in speech recognition, as a way of generating pronunciations for new words that may be available in grapheme form, or for naive users to add new words more easily. In that case, the system must generate the multiple variations of the word. In recent works, we noticed (Schlippe et al., 2012) that created Grapheme-To-Phoneme models for Indo-European languages with word-pronunciation pairs from the GlobalPhone project and from Wiktionary and tested for Czech, English, French, Spanish, Polish, and German ASR. Wiktionary pronunciations have been provided by the Internet community and can be used to quickly and economically create pronunciation dictionaries for new languages and domains. An other solution was proposed in (Laurent et al., 2009), where the Grapheme-To-Phoneme system uses statistical machine translation techniques. The generated word pronunciations are employed in the dictionary of the ASR system.

2.3. Under-resourced languages

There are more than 6000 languages in the world but only a small number possess the resources required for implementation of Human Language Technologies (HLT). Thus, HLT are mostly concerned by languages which have large resources available or which suddenly became of interest because of the economic or political scene. On the contrary, languages from developing countries or minorities were less treated in the past years. Among HLT, phonetization is also concerned about this fact: less-resourced languages are also investigated since the 2000s. It is not possible to make an exhaustive review, but we noticed the followings: for Malay (El-Imam and Don, 2000), for Thai (Tarsaku et al., 2001), for Korean (Kim et al., 2002), for Punjabi (Gera, 2006), for Romanian (József et al., 2011), for Arabic (El-Imam, 2004), for Greek (Chalamandaris et al., 2005) or for Polish (Demenko et al., 2003). In all these studies, authors adopted various solutions mainly depending on the availability of resources and on the structural of the language.

It is also important to mention that in some languages, code-switching is a common practice and the phonetization system can be face on such a phenomena. Some specific strategies can be adopted, as proposed in (Thangthai et al., 2007).

3. Phonetization approach for FA

3.1. Overview

The “Forced Alignment”' task included both phonetization and alignment tasks. Phonetization is the process of representing sounds by phonetic signs. Alignment is the process of aligning speech with these sounds.

To our knowledge, only one public FA system includes a rule-based phonetization step; this system is described in (Goldman, 2011). The grapheme conversion tool is provided by an external TTS system and suggests some pronunciation variants. The optional phonemes are marked as an expert annotator can compare the sequence of phonetic symbols with the audible speech of each utterance and select the most appropriate. This approach is well suited for read speech, but we suppose that manual corrections must be applied in case of spontaneous speech. Moreover, a new Letter-To-Sound system must be entirely developed to handle each new language.

In many forced-alignment systems based on ASR technologies, the phonetization step is limited to a sequence of dictionary look-ups. The dictionary contains words with a set of pronunciations (the canonical one, and optionally some common reductions, etc). Phonetization is then proposed for the aligner to choose the phoneme string *because the pronunciation generally can be observed in the speech.* In this approach, it is then assumed that all words of the speech transcription and their phonetic variants are mentioned in the pronunciation dictionary. Fortunately, a large set of these instances can be extracted from a lexicon of systematic variants even if it will not cover all the possible observed realizations. Moreover, with time, computer memory is becoming ever cheaper, then larger and better dictionaries are now available for many languages. Accordingly, it could be argued that the importance of some kind of ”back-up” strategy is declining. Although 1/ it is of course true for the couple (computers, major-languages) but this argument can be less important for an under-resourced language and 2/ the more pronunciations are added, the more confusion may occur for the aligner.

The solution we propose aims to combine the advantages of the various approaches and can be applied to a large set of languages. Firstly, we choose a knowledge-based approach, as data-driven approaches requires a large set of data for the training stage and such a data are not always available (particularly for less-resourced languages). We did not introduced specific rules in the system, in order that the system is language-independent (only the given resources are language-specific). Moreover, our approach does not depend on the writing system (it works indifferently on French or Chinese).

In spontaneous speech, many phonetic variations occur. Some of these phonologically known variants are predictable and can be included in the pronunciation dictionary but many others are still unpredictable (especially invented words, regional words or words borrowed from another language.). The process of transcribing text into sounds starts by pre-processing the text and representing it by lexical items to which the phonetization are applicable. In our system, we use the multilingual text normalization approach proposed in (Bigi, 2011).

3.2. Phonetization based on resources

As in ASR systems, we choose the dictionary based solution, which consist in storing a phonological knowledge in a lexicon. In this sense, this approach is *language-independent* unlike rule-based systems. The dictionary includes phonetic variants that are proposed for the aligner to choose the phoneme string. The hypothesis is that the answer to the phonetization question is in the signal.

An important step is to build the pronunciation dictionary, where each word in the vocabulary is expanded into its constituent phones. For example, the French sentence ”je suis” (*I am*) can be:

- /ʒsɥi/ is the standard pronunciation,
- /ʒsɥiz/ is the standard pronunciation plus a liaison,
- /ʒəsɥi/ is the South of France pronunciation,
- /ʒəsɥiz/ is the previous pronunciation plus a liaison,
- /ʃɥi/ is a very frequent specific realization.

The dictionary entries for both words are:

je [je] ʒ	suis [suis] sɥi
je(2) [je] ʒə	suis(2) [suis] sɥiz
je(3) [je] ʃ	suis(3) [suis] sui
	suis(4) [suis] ɥi
	suis(5) [suis] ɥiz

Depending on the language, the availability of resources is different. In our data set, the dictionary includes a large set of entries (English, French, Italian), an acceptable number of entries (Chinese, Vietnamese) or a poor number of entries (Taiwan Southern Min). The dictionary file format used is HTK-standard (Young and Young, 1994). The English dictionary was downloaded from the CMU and was not modified. The French and the Italian dictionaries were created by merging available TTS systems dictionaries and ASR systems dictionaries. They was also enriched by word pronunciations observed in spontaneous speech corpora. We corrected manually a large set of these both phonetizations. For example, the Italian dictionary contains a set of possible pronunciations of words, including accents as *perchè* pronounced as [b e r k e], and reduction phenomena as [p e k] (or [k wa] for the word *acqua*). The Chinese dictionary is made of characters and of the most frequent words. It allows users to add phonetic variants for these frequent entries. The Taiwanese dictionary is made of a small set of syllables, written in pinyin.

3.3. Phonetization algorithm

As in TTS systems, a specific algorithm to phonetize entries was also developed. As the data-driven approaches, our grapheme-to-phoneme conversion system is based on the idea that given enough examples it should be possible to predict the pronunciation of unseen words purely by analogy. Unlike these approaches, our system is then applied to missing words during the phonetization process (and not during a training stage), based on knowledge provided by the dictionary.

The algorithm consists in exploring the unknown entry from left to right and to find the longest strings in the dictionary. Since this algorithm uses the dictionary, the quality of such a phonetization will depend on this resource. The algorithm is described in the following pseudo-code:

```
function phonetize(input:word):

    if len(word) == 0 then
        return ""
    fi

    # Find the longest left string that can
    # be phonetized from the dictionary
    left = get_longest_part(word)
    phonleft = get_in_pronunciationdict(left)
    if len(left) == len(word) then
        return phonleft
    fi

    # Find how to phonetize right part
    # Get the right un-phonetized subpart
    right = subpart(word)
    if len(right) == 0 then
        return phonleft
    fi
    phonright = get_in_pronunciationdict(right)
    if phonright is None then
        phonright = phonetize(right)
    fi

    return concatenate(phonleft,phonright)
```

The algorithm is currently described and implemented from left to right, but it can be easily transposed from right to left.

One difficulty by applying this algorithm is due to phonetic variants. Actually, the function `get_in_pronunciationdict()` applied to any string sequence return all available pronunciations of this entry. For example, if this algorithm is applied to the string ”jesuis”, with our French dictionary, the result will contains all variants described previously: ʒ|ʒə|ʃ sɥi|sɥiz|sui|ɥi|ɥiz where pipes separates variants and the white space separates left/right parts. For a sake of simplicity, this result is stored into a DAG - a Directed acyclic graph (Figure 1).

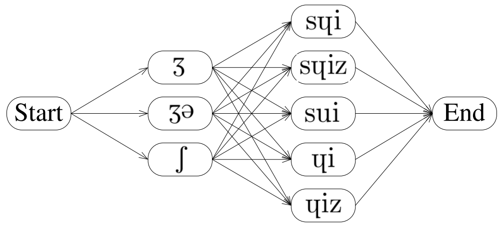

Fig. 1: DAG with phonetic variants

The final pronunciations are extracted by exploring all paths of this DAG. As we can see, the number of variants can significantly increase. That's the reason why, we introduced the possibility to get only a limited number of variants. We choose to select the shortest ones (i.e. the fewest number of nodes), which is a reasonable solution due to a larger number of speech reductions than speech over-production!

4. Results

4.1. Software

The algorithm and resources described in this paper are integrated in the SPPAS tool (Bigi, 2012), a tool to automatically produce annotations which include utterance, word, syllabic and phonemic segmentation from a recorded speech sound and its transcription. It is currently designed for French, English, Italian, Chinese and Taiwanese. SPPAS is open source software issued under the GNU Public License. and is multi-platform (Linux, MacOS and Windows). Moreover, SPPAS is specifically designed to be used directly by linguists.

4.2. Phonetization of unknown words

The experiments were carried out on French because all required resources were freely available: the dictionary and the test corpus. The dictionary is available in the SPPAS tool: it is made of 348k tokens, and it includes a large set of variants: regional variants, the liaison phenomena and common reduction phonemena. The Marc-FR corpus was used as test corpus (Bigi et al., 2012). This corpus is based on parts of three different French corpora and was downloaded from the SLDR - Speech & Language Data Repository, at:

http://www.sldr.fr/sldr000786/fr

About two minutes of 3 different corpora (7 minutes altogether) were manually segmented and transcribed: read speech from the AixOx Corpus (Herment et al., 2012), conversational speech from from CID - Corpus of Interactional Data (Blache et al., 2010), and a political discourse (Bigi et al., 2011). The whole corpus is made of 1220 tokens, 5400 phonemes.

The phonetization system was launched on the Marc-FR corpus, by using the whole French dictionary (650k). The results are as follow:

- 1175 tokens are in the dictionary and the manual phonetization is proposed;
- 13 tokens are in the dictionary but the manual phonetization is not proposed (i.e. 1,07%);
- 32 tokens are not in the dictionary (i.e. 2.62% of the tokens).

This result confirms that even with a very large dictionary, a quite significant number of phonetization (or variants) are missing (3.69%). The list of unknown tokens consists in 3 proper names and 29 reductions or mispronunciations, distributed as:

- 6 in the read speech,
- 2 in the political discourse,
- 21 in the conversational corpus.

As expected, missing entries are mainly coming from spontaneous speech. The proposed algorithm is then used to phonetize these tokens.

If the number of variants is limited to 4, 22 tokens are phonetized properly (i.e. 69%). While the number of variants is extended to 8, 26 tokens are phonetized properly (i.e. 81%).

5. Conclusion

This paper presented a phonetization system entirely designed to handle multiple languages and/or tasks with the same algorithms and the same tools. Only resources are language-specific, and the approach is based on the simplest resources as possible. Next work will consist to reduce the number of entries in the current dictionaries. Indeed, all tokens that can be phonetized properly by our algorithm could be removed of the dictionary. Hence, we hope this work will be helpful in the future to open to new practices in the methodology and tool developments: thinking problems with a generic multilingual aspect, and distribute tools with a public license.

6. Acknowledgement

This research was partially supported by the Ortolang project. The support is gratefully acknowledged. http://www.ortolang.fr. We address special thanks to Tatsuya Watanabe for the help in the python development.

References

Allen, J., Hunnicutt, M. S., and Dennis, H. (1987). *From Text to Speech: The MIT talk System*. Cambridge University Press.

Belrhali, R., Aubergé, V., and Boë, L.-J. (1992). From lexicon to rules: toward a descriptive method of french text-to-phonetics transcription. In *The Second International Conference on Spoken Language Processing*.

Bigi, B. (2011). A multilingual text normalization approach. In *2nd Less-Resourced Languages workshop, 5th Language & Technology Conference*, Poznan, Poland.

Bigi, B. (2012). SPPAS: a tool for the phonetic segmentations of Speech. In *The eighth international conference on Language Resources and Evaluation, ISBN 978-2-9517408-7-7*, pages 1748–1755, Istanbul, Turkey.

Bigi, B., Péri, P., and Bertrand, R. (2012). Orthographic Transcription: Which Enrichment is required for Phonetization? In *The eighth international conference on Language Resources and Evaluation, ISBN 978-2-9517408-7-7*, pages 1756–1763, Istanbul, Turkey.

Bigi, B., Portes, C., Steuckardt, A., and Tellier, M. (2011). Multimodal annotations and categorization for political debates. In *ICMI Workshop on Multimodal Corpora for Machine learning*, Alicante (Spain).

Bisani, M. and Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451.

Blache, P., Bertrand, R., Bigi, B., Bruno, E., Cela, E., Espesser, R., Ferré, G., Guardiola, M., Hirst, D., Magro, E.-P., Martin, J.-C., Meunier, C., Morel, M.-A., Murisasco, E., Nesterenko, I., Nocera, P., Pallaud, B., Prévot, L., Priego-Valverde, B., Seinturier, J., Tan, N., Tellier, M., and Rauzy, S. (2010). Multimodal annotation of conversational data. In *The Fourth Linguistic Annotation Workshop*, pages 186–191, Uppsala, Sueden.

Caseiro, D., Trancoso, L., Oliveira, L., and Viana, C. (2002). Grapheme-to-phone using finite-state transducers. In *IEEE Workshop on Speech Synthesis*, pages 215–218.

Chalamandaris, A., Raptis, S., and Tsiakoulis, P. (2005). Rule-based grapheme-to-phoneme method for the greek. *trees*, 18:19.

Daelemans, W. and Van Den Bosch, A. (1997). Languageindependent data-oriented grapheme-to-phoneme conversion. *Progress in speech synthesis*, pages 77–89.

Damper, R., Marchand, Y., Adamson, M., and Gustafson, K. (1998). Comparative evaluation of letter-to-sound conversion techniques for english text-to-speech synthesis. In *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*.

Demenko, G., Wypych, M., and Baranowska, E. (2003). Implementation of grapheme-to-phoneme rules and extended sampa alphabet in polish text-to-speech synthesis. *Speech and Language Technology*, 7:79–97.

Divay, M. and Guyomard, M. (1977). Grapheme-to-phoneme transcription for french. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 575–578.

Dutoit, T. (1997). *An introduction to text to speech synthesis*, volume 3. Springer.

El-Imam, Y. (2004). Phonetization of arabic: rules and algorithms. *Computer Speech & Language*, 18(4):339–373.

El-Imam, Y. and Don, Z. (2000). Text-to-speech conversion of standard malay. *International Journal of Speech Technology*, 3(2):129–146.

Galescu, L. and Allen, J. (2001). Bi-directional conversion between graphemes and phonemes using a joint n-gram model. In *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*.

Gera, P. (2006). Text to speech synthesis for punjabi language.

Goldman, J.-P. (2011). EasyAlign: a friendly automatic phonetic alignment tool under Praat. In *Interspeech*, number Ses1-S3:2, Florence, Italy.

Herment, S., Loukina, A., Tortel, A., Hirst, D., and Bigi, B. (2012). A multi-layered learners corpus: automatic annotation. In *4th International conference on corpus linguistics Language, corpora and applications: diversity and change*, Jaén (Spain).

Jiampojamarn, S., Cherry, C., and Kondrak, G. (2008). Joint processing and discriminative training for letter-to-phoneme conversion. In *ACL*, pages 905–913.

József, D., Ovidiu, B., and Gavril, T. (2011). Automated grapheme-to-phoneme conversion system for romanian. In *6th Conference on Speech Technology and Human-Computer Dialogue*, pages 1–6.

Kim, B., Lee, G. G., and Lee, J.-H. (2002). Morpheme-based grapheme to phoneme conversion using phonetic patterns and morphophonemic connectivity information. *Journal ACM Transactions on Asian Language Information Processing*, 1(1):65–82.

Laurent, A., Deléglise, P., and Meignier, S. (2009). Grapheme to phoneme conversion using an smt system. In *Interspeech*, pages 708–711.

Levinson, S., Olive, J., and Tschirgi, J. (1993). Speech synthesis in telecommunications. *Communications Magazine, IEEE*, 31(11):46–53.

Schlippe, T., Ochs, S., and Schultz, T. (2012). Grapheme-to-phoneme model generation for indo-european languages. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4801–4804.

Tarsaku, P., Sornlertlamvanich, V., and Thongprasirt, R. (2001). Thai grapheme-to-phoneme using probabilistic GLR parser. In *Interspeech*, Aalborg, Denmark.

Taylor, P. (2005). Hidden markov models for grapheme to phoneme conversion. In *Interspeech*, pages 1973–1976.

Thangthai, A., Wutiwiwatchai, C., Rugchatjaroen, A., and Saychum, S. (2007). A learning method for thai phonetization of english words. In *Interspeech*, pages 1777–1780.

Torkkola, K. (1993). An efficient way to learn english grapheme-to-phoneme rules automatically. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 199–202.

Young, S. and Young, S. (1994). The htk hidden markov model toolkit: Design and philosophy. *Entropic Cambridge Research Laboratory, Ltd*, 2:2–44.

Yvon, F., de Mareüil, P. B., et al. (1998). Objective evaluation of grapheme to phoneme conversion for text-to-speech synthesis in french. *Computer Speech & Language*, 12(4):393–410.