Generation of Word Alternative Pronunciations Using Weighted Finite State Transducers

Sérgio Paulo, Luís C. Oliveira

L^2F Spoken Language Systems Lab.
INESC-ID/IST
Rua Alves Redol 9, 1000-029 Lisbon, Portugal
{spaulo,lco}@l2f.inesc-id.pt

Abstract

This paper describes a speech segmentation tool allowing alternative word pronunciations within a WFST framework. Two approaches to word pronunciation graph generation were developed and evaluated. The first approach is grapheme-based where each grapheme is converted into all the phones it can give rise to, in the form of a WFST. Word graphs are obtained by concatenating all grapheme WFSTs. In the second approach, a training corpus is used to find the different realizations of the syllable. This information is used to generate alternative syllable-level pronunciations, represented as WFSTs, that are concatenated to produce the word graphs. Both approaches were evaluated by aligning the phone sequence generated by each approach with the manually labelled phone sequence for all utterance in the corpus. This alignment was used for computing F-rate values for each phone. The syllable-based approach produced the best results.

1. Introduction

Data driven approaches have been applied successfully to speech synthesis (corpus-based speech synthesizers[1],[2],[3]), natural prosody generation and speech recognition (HMM-based speech recognizers). However, the quality of the resulting systems depends on the quality of corpora segmentations (e.g. badly annotated phones can cause a concatenative TTS to produce a low quality signal). While the ability to produce accurate speech sounds in rapid succession is something we humans take for granted, it is a complex cognitive task involving a hierarchical structure of execution that extends from the production of syntactically and semantically organized sentences or phrases down to the production of phones. Thus, the accurate prediction of the speaker's acoustic realization of the text prompts is a very difficult task. The growing size of speech research corpora makes manual segmentation unfeasible and automatic approaches have been developed to carry out this task. The automatic tools must be flexible enough to deal with the speaker-specific word pronunciations. In [4], an iterative approach to detect the speaker's word pronunciations is proposed and a higher quality speech signal was achieved by using the speaker-specific word pronunciation rather than the pronunciation predicted by the system's lexicon. Similar conclusions were drawn in [5].

Several corpus-based speech synthesizers have been developed under the *WFST* framework in the last few years, e.g. [6] and [7]. This framework is well suited to perform a flexible unit-selection, taking advantage of the multiple acceptable ways an utterance can be produced, so that the quality of the resulting signal can be improved. The flexible utterance representation allowed by *WFST*s led us to the development of a *WFST*-based segmentation tool allowing alternative pronunciations of words.

In this paper, we present the *WFST*-based segmentation tool (section 2) and two new approaches for generating the alternative pronunciations (section 3). A comparison between the segmentation results of both approaches is shown in section 4.

2. Phonetic Segmentation Tool

The phonetic segmentation tool described in this section consists of an HMM-based segmenter using the *WFST* framework. Such a tool computes the utterance segmentation by finding the best path of a *WFST* that converts a sequence of frame indexes into a series of phonetic segments. This *WFST* is the result of the composition of three *WFST*s that represent the utterances at three different levels. The first one ($WFST_{frame}$, Fig. 1) converts the signal frame indexes into HMM states. Each edge of the $WFST_{frame}$ is weighted by the symmetric of the logarithmic value of the probability that that feature vector is generated given a certain phone state. Observing Fig. 1, two further edges can be seen. Each one of these edges has the same symbol for both input and output labels. Label end is used for marking the end of each phone, and label $jump$ is used to allow the system to delete some phones of the utterance, as can be seen later on in this paper. The second *WFST* ($WFST_{state}$, Fig. 2) converts the phone states into phones. It consists of the union of every phone model, and its edges are weighted by minus logarithm of the transition probability from a given state to another state within the same phone model. The last transducer (*WFST*$_{phone}$, Fig. 3) represents the utterance with multiple word pronunciations as well as cross-word co-articulations, as will be seen later on in section 3.

All the *WFST*-related operations described in this paper were performed by using a library developed at our lab based on the Edinburgh Speech Tools([10]). Many new *WFST* operations are provided by the library, like the on-the-fly *WFST* composition, for example.

2.1. Phone model training

The proposed segmentation tool is mainly targeted for the segmentation of large single-speaker corpora, and when the speaker phone models are not available, new ones have to be trained. The training procedure starts by using a DTW-based segmentation tool ([8]) to generate the model training data. Then, the HMM ToolKit (HTK,[9]) is used to create phone models based on the segmentation performed before. After generating the models, the corpus is segmented again, by using the models created in the previous step. The training procedure continues by re-training the models until they converge or until a maximum number of iterations is achieved. The resulting models are then converted from the HTK format to a format suitable to our own segmenter.

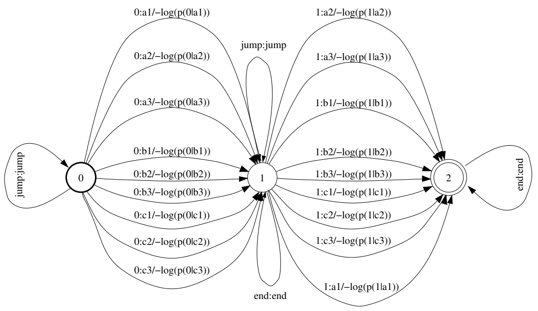

Figure 1: *WFST that converts frame indexes into phone states.*

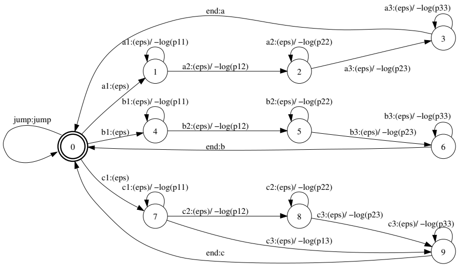

Figure 2: *WFST that converts phone states into phones.*

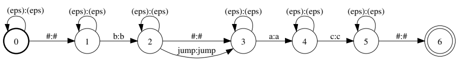

Figure 3: *WFST that represents the utterance word pronunciations in terms of phones.*

3. Utterance representation with *WFST*s

The speaker's acoustic realization of the text prompts is not always easy to predict. This problem is even more difficult if the recordings were made by non-professional speakers. Variations can occur either at the prosodic level or at the phonetic level, or at both the levels. For instance, major word breaks can be set at different positions within the text, as [11] has shown. At the phonetic level, such variations arise as variations on word pronunciations. The way a word is uttered by a speaker depends on multiple factors, and an automatic tool to perform speech segmentation at the phonetic level must deal, in some extent, with flexible ways of representing words as well as whole utterances. Flexibility on utterance representation is needed for the segmentation tool to choose the most likely representation based on the speech signal acoustic features. However, if too much flexibility is allowed, one can fall in the case of purely acoustically-based ways of choosing the word pronunciations. Such a system can perform poorly if higher level and more reliable information is not used. In [4], a phone recognizer is used together with an n-gram grammar of phones to achieve reliable corpus transcriptions based on the signal acoustics with an iterative algorithm. In that work, the authors noticed, too, that sometimes undesired phones are inserted in the middle of some particular phone pairs, which implies that some post-processing is needed for minimizing the *noise* generated by this approach. We propose two approaches to the creation of alternative utterance transcriptions in a much more controlled way.

3.1. Grapheme-based representation

In the first approach, the grapheme is used as the atomic unit for creating the multiple word pronunciations. By inspecting the grapheme-to-phone rules of the language (the current work deals with European Portuguese) each grapheme is assigned a set of possible phones that it can give rise to. This information is hold in a *WFST* receiving graphemes as inputs and producing the associated phones (let it be the $WFST_{GtoP}$). For each word, a grapheme *WFST* is built using the list of the word graphemes (let it be $WFST_{grapheme}$), as is depicted in Fig. 4.

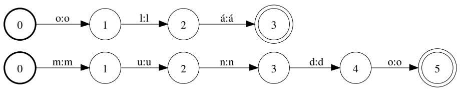

Figure 4: *WFSTs representing the graphemes of the words ”olá” and ”mundo”.*

The $WFST_{Pron}$, resulting from the composition of $WFST_{grapheme}$ with $WFST_{GtoP}$, represents the multiple possible ways of pronouncing the word under analysis ($WFST_{Pron} = WFST_{grapheme} \circ WFST_{GtoP}$). By concatenating all word *WFST*s with an optional silence *WFST*, a flexible representation for a given utterance can be achieved. Fig. 5 shows the representation of the sentence *”Olá mundo.”* (”Hello world.”). In that figure, the utterance graph was split into three parts so that the labels can be more easily read.

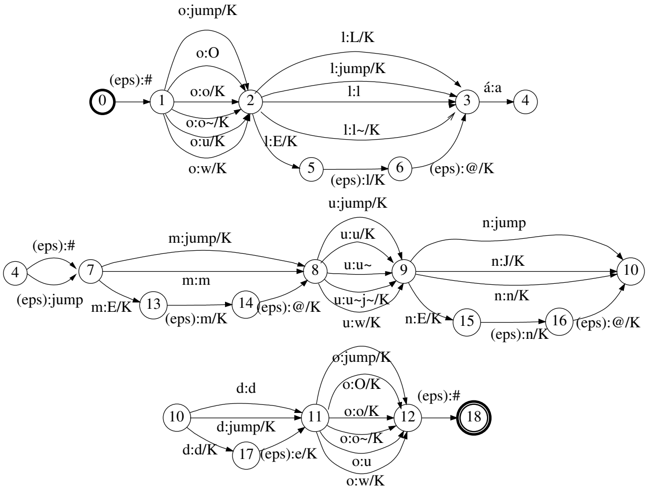

Figure 5: *Grapheme-based WFST representation of the sentence ”Olá mundo”.*

Regarding the Fig. 5, some issues must be pointed out:

- When the edge output label is ”jump”, no phone is associated with the grapheme;
- When the edge input is ”*(eps)*”, it means that a grapheme has given rise to more than one phone (except for the case of the silence, #);
- When the edges have labels according to the canonical pronunciation of the word, they have zero cost, otherwise their costs are equal to the cost of choosing an alternative pronunciation (K).

By projecting the output labels of the $WFST_{Pron}$ onto the inputs, we create a transducer ($WFST_{utt\_isol}$) with a format suitable to the segmenter.

3.2. Syllable-based representation

By inspecting the large number of possibilities that the former approach could generate, and fearing that it could harm the segmentation performance, a more controlled approach was developed. This approach consists of two steps:

- collection of alternative pronunciations for each syllable;
- generation of alternative word pronunciations.

3.2.1. Collection of syllable's alternative pronunciations

Sixty thousand words were selected from a text corpus of Portuguese newspapers. The phonetic transcription of these words, were carried out by using a set of grapheme-to-phone rules together with an exception dictionary[1]. Then, the alignment between the grapheme and phone sequences of each word was performed by minimizing the string-edit distance between grapheme and phone strings. Once the phone-based syllabification is known, it can be mapped into the grapheme level by using the grapheme-phone alignment. At this point we can set the *grapheme-level syllables* ($GrafSyl$). For example: let *"mundo"* ("world") be a word instance. Its canonical transcription is *"(m u˜ d u)"*[2]. The alignment between graphemes and phones results in: {(G=m, P=m);(G=u, P=u˜);(G=n, P=$jump$);(G=d, P=d);(G=o, P=u)}, where (G=n, P=$jump$) means that the grapheme m did not give rise to any phone. The application of syllabification rules to this word phones results in *"((m u˜); (d u))"*. Then, phonetic syllabification is mapped into the grapheme level by using the alignment previously described. The resulting *grapheme-level syllables* are *("mun"; "do")*. Since we know which phones are associated with each *grapheme-level syllable* ("mun" is associated with *"(m u˜)"* and "do" is associated with *"(d u)"*), all the different pronunciations of each *grapheme-level syllable* that are observed within the lexicon can be collected and associated with the respective *grapheme-level syllable*, in order to be used for word pronunciation generation. Given that the *grapheme-level syllable* conversion into phones depends on its context, additional features were used to distinguish the multiple versions of the same grapheme-level syllable:

St: Syllable stress;

WI: A binary feature to indicate whether the syllable is word initial ($value = 1$) or not ($value = 0$);

WF: A binary feature to indicate whether the syllable is word final ($value = 1$) or not ($value = 0$)

NV: A binary feature to indicate whether the syllable is followed by a vowel ($value = 1$) or not ($value = 0$);

PV: A binary feature to indicate whether the syllable is preceded by a vowel ($value = 1$) or not ($value = 0$).

The name of the context-dependent $GrafSyl$ is assigned by using the rule: $GrafSyl\_St\_WI\_WF\_NV\_PV$. Recalling the previous example, "mun_1_1_0_0_0" and "do_0_0_1_0_1" are the names of context-dependent $GrafSyl$s "mun" and "do", respectively. By using the context-dependent $GrafSyl$s as atomic units for computing the alternative pronunciation, many alternatives that do not make sense in that context can be avoided.

All context-dependent $GrafSyl$, as well as their multiple pronunciations, are stored so that they can later be used for creating new word pronunciations.

3.2.2. Generation of alternative word pronunciations

In order to generate alternative pronunciations for a word, a series of steps must be carried out: firstly, the word canonical transcription and the phone syllabification must be determined; secondly, the alignment between word graphemes and phones is computed, which allows us to make the context-dependent grapheme-level syllables of the word. The following step is the search for alternative pronunciations for the syllable within the data collected earlier. If any alternative pronunciation is found for a given syllable, a new path[3] is added to the *WFST* that represents that syllable pronunciation. The alternative paths are weighted by fixed value K. The impact of the value of K on the performance of the segmenter will be shown later. The multiple word pronunciations are encoded in a *WFST* resulting from the concatenation of all the syllable *WFST*s. Finally, the utterance *WFST* (let it be $WFST_{utt\_isol}$) consists of the concatenation of all word *WFST*s with an optional silence between any two words (it can be seen on the transitions from the state 4 to state 5 in Fig. 6). Fig. 6 shows the graph generated by this approach for the same example sentence used earlier in this section. By inspecting that figure, it can be noticed that much less alternatives are produced by this technique.

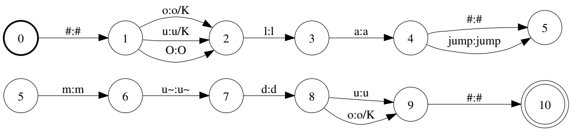

Figure 6: *Syllable-based WFST representation of the sentence "Olá mundo".*

3.3. Cross-word co-articulations

In the previous subsection, two different approaches to generate alternative word pronunciations were presented. In order to take into account the cross-word co-articulations, a *WFST* encoding the language-specific post-lexical rules is composed with the $WFST_{utt\_isol}$. The *WFST* resulting from this operation is provided to the segmentation tool.

4. Evaluation

This section aims to evaluate four different approaches for segmenting an utterance. The basic approach is a segmenter that only allows the canonical word pronunciation (System A). The second approach, a bit more elaborated, consists in locating the inter-word silences, and applying the cross-word co-articulation rules between any two words that are not separated by a silence (System B). None of the former approaches allows multiple pronunciations per word. The third approach consists of adding a set of cross-word co-articulation alternatives to the isolated word graph generated by using the Grapheme-based approach (system C). The forth approach is similar to the third, but in this case the isolated word graph is generated by using the Syllable-based approach (system D). The evaluation was performed by dividing the phone set into three groups, the vowels, the voiced consonants, and the unvoiced consonants. Values of precision and recall were computed for each phone of the three groups for evaluating the accuracy of the segmentation tool on the detection of those phones. The F-rate (the harmonic mean of the precision and recall values[12]) was used as performance measure.

[1] At this point, one get the phones split into syllables, too, by applying a set of syllabification rules.

[2] The phonetic transcriptions that are shown throughout this text use the SAMPA phone set for European Portuguese, for further information refer to http://www.l2f.inesc-id.pt/˜lco/ptsam/ptsam.pdf

[3] For each alternative pronunciation.

Table 1: *Accuracy of the different segmentation systems for detection of Vowels (F-Measure in %).*

	Alternative pronunciation cost (K)				
$System$	0	20	80	150	300
A	86.4	86.4	86.4	86.4	86.4
B	94.6	94.6	94.6	94.6	94.6
C	55.2	92.6	95.4	95.3	95.2
D	95.4	96.0	96.2	96.0	95.8

Table 2: *Accuracy of the different segmentation systems for detection of Voiced Consonants (F-Measure in %).*

	Alternative pronunciation cost (K)				
$System$	0	20	80	150	300
A	95.6	95.6	95.6	95.6	95.6
B	97.5	97.5	97.5	97.5	97.5
C	61.2	97.7	99.0	98.9	98.7
D	98.8	98.9	99.1	99.1	99.0

The first conclusion that can be drawn from the analysis of the results shown in Tables 1, 2 and 3, is that the speaker's acoustic realization of the text prompts is very closed to what was predicted by the system's lexicon. This was not a surprise, given that the prompts were read by a professional speaker. A second comment is that the system C performs poorly when alternative pronunciations are not weighted by any additional cost ($K = 0$). This is what we expected the system C to do, because it allows a large number of alternative pronunciations, without penalizing the non-canonical ones. The variation of the systems' accuracy with the value of the cost K, suggests that a good value for K is somewhere between $K = 80$ and $K = 150$. Both the new approaches outperform the ones that do not allow alternative pronunciations, except for Unvoiced consonants. In this case, the post-lexical rules of system B outperform all other systems. However, systems C and D are still better than system A. The best performance of the system B for detecting the unvoiced consonants at the word boundaries can be explained by two main reasons: the speaker is very regular in terms of cross-word co-articulations, and the models for unvoiced consonants still need adaptation. This problem can be overcome by re-training new models over the new segmentation results. After re-training the models and performing the segmentation again, the syllable-based approach (with $K = 80$) generates F-rate values of 96.3%, 99.1% and 99.0%, for vowels, voiced consonants and unvoiced consonants, respectively.

5. Conclusions

In this work a WFST-based phonetic segmentation tool, using word graphs for producing alternative pronunciations, was presented. Two techniques to generate these word graphs were presented: a grapheme-based one and a syllable-based one. The performance of the segmentation tool with alternative word pronunciation was compared with the results obtained by using a canonical transcription with and without cross-word co-articulation rules. By using an appropriate weighting value for the alternative pronunciations, the new techniques outperformed the traditional ones. The best results were achieved using the word graph produced by the syllable-based technique.

Table 3: *Accuracy of the different segmentation systems for detection of Unvoiced Consonants (F-Measure in %).*

	Alternative pronunciation cost (K)				
$System$	0	20	80	150	300
A	94.7	94.7	94.7	94.7	94.7
B	99.1	99.1	99.1	99.1	99.1
C	60.5	97.3	98.7	98.3	98.1
D	98.6	98.6	98.7	98.7	98.6

6. Acknowledgements

This work is part of Sérgio Paulo's PhD Thesis sponsored by a Portuguese Foundation for Science and Technology (FCT) scholarship, and was partially funded by FCT project POSI/PLP/47175/2002.

7. References

[1] A. Hunt and A. Black, *Unit selection in a concatenative speech synthesis system using a large speech database.* In Proceedings of ICASSP 96, Atlanta, Georgia, USA, 1996.

[2] A. Black and P. Taylor, *CHATR: a generic speech synthesis system.* In Proceedings of COLING-94, Kyoto, Japan, 1994.

[3] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou and A. Syrdal, *The AT&T Next-Gen TTS System.* 137^{th} Acoustical Society of America meeting, Berlin, Germany, 1999.

[4] Y. Kim, A. Syrdal and M. Jilka, *Improving TTS by Higher Agreement Between Predicted Versus Observed Pronunciations.* In Proceedings of 5th ISCA Speech Synthesis Workshop, Pittsburgh, USA, June 2004.

[5] J. Fackrell, W. Skut and K. Hammervold, *Improving the accuracy of pronunciation prediction for unit selection TTS.* In Proceedings of Eurospeech 2003, Geneva, Switzerland, September 2003.

[6] J. Yi, J. Glass and L. Hetherington, *A Flexible, Scalable Finite-State Transducer Architecture for Corpus-Based Concatenative Speech Synthesis.* In Proceedings of ICSLP 2000, Beijing, China, October 2000.

[7] I. Bulyko and M. Ostendorf, *Efficient Integrated Response Generation from Multiple Targets using Weighted Finite-State Transducers.* Computer Speech and Language, Vol. 16, no. 3/4, pp. 533-550, 2002.

[8] S. Paulo and L. C. Oliveira, *DTW-based Phonetic Alignment Using Multiple Acoustic Features.* In Proceedings of Eurospeech 2003, Geneva, Switzerland, September 2003.

[9] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev and Phil Woodland, *The HTK Book*.

[10] P. Taylor, R. Caley, A. Black and S. King, *Edinburgh Speech Tools Library* System Documentation Edition 1.2, 15th June 1999.

[11] M. C. Viana, L. C. Oliveira and A. I. Mata, *Prosodic Phrasing: Machine and Human Evaluation*, In Proceedings of 4th ISCA Speech Synthesis Workshop, Scotland, September 2001.

[12] C. J. van Rijsbergen, *Information Retrieval*. Butterworhs, London, 1979.