PHONETIC GRAMMARS FOR GF

JANNA KHEGAI

Abstract. We discuss the problem of using well-known Festival speech synthesis system for languages with no voices/lexicons provided in the current distributions. We consider various parts of the system implemented according to the standard Text-To-Speech (TTS) system architecture.

1. Grammatical Framework - grammar formalism for multilingual translation

At Chalmers University of Technology I am working within Grammatical Framework project (GF)[6] implemented in Computer Science department in the functional programming language Haskell. GF is a multilingual translation system of INTERLINGUA type. It implements the language for expressing grammars using formalism based on type theory [7]. The main feature of GF grammar formalism is explicit separation of abstract syntax shared between different languages and parallel concrete syntaxes for each language. Both parts have a type systems, that allows to verify input well-formedness as well as resolve ambiguities using semantic information contained in object type. GF could be used for defining formal and natural languages. So far, GF grammars have been written for fragments of about 20 natural languages. As for any system with such a general purpose as natural language processing convenient user-friendly interface is very important. Although GF is intended for translation of texts it would be nice to have a component for TTS generation. Since TTS is a non-trivial problem [4] , we would like to use some existing synthesis library more or less like a black-box for rendering text as speech. In fact, we used the Festival speech-synthesis system for saying sentences in some languages included in the distribution [1]. In this document we will discuss how Festival could be used for languages not yet implemented in Festival.

2. Festival speech synthesis system

Within Festival we can identify three basic parts of the TTS process:

- Text analysis: From raw text to identified words and basic utterances.
- Linguistic analysis: Finding pronunciations of the words and assigning prosodic structure to them: phrasing, intonation and durations.
- Waveform generation: From a fully specified form (pronunciation and prosody) generate a waveform.

Festival is written in C++. It also includes a full programming language, Scheme (a variant of the functional programming language Lisp) as a powerful interface to speech synthesis functions. It is Scheme scripts that control various stages in speech processing in order to adjust the system for the current needs.

3. Linguistic/text analysis

We can represent the pronunciations of words in terms of units called phones. The standard system for representing phones is the International Phonetic Alphabet (IPA). Phones can be described by how they are produced articulatorily by the vocal organs. Consonants are defined in terms of their place and manner of articulation and voicing, vowels - by their height and backness. A phoneme is a generalization or abstraction over different phonetic realizations. Allophonic rules express how a phoneme is realized in a given context. Pronunciation dictionaries gives the pronunciation of words as strings of phones, sometimes including syllabification and stress. Most online pronunciation dictionaries have on the order of 100000 words, but still lack many names, acronyms and inflected forms. The text analysis component of a TTS system maps from orthography to strings of phones. This is usually done with a large lexicon augmented with a system for handling productive morphology, pronunciation changes, names, numbers and acronyms.

It is the lexicons job to produce a pronunciation of a given word. In general TTS systems, it is not possible to guarantee that a lexicon will contain all the words found in a text, therefore some system for predicting pronunciation from the word itself is necessary. Three the most important cases where we cannot rely on a word dictionary involve names, morphological productivity, and numbers. Even when a large lexicon can be constructed to cover the whole vocabulary it would be useful to find a principled method to reduce the size of the lexicon.

In many languages the orthographic system has some relationship to the pronunciation, depending on the language it may be trivial (such as in Spanish) or relatively difficult (English), or harder (Japanese). Humans can often pronounce words reasonably even when they have never seen them before. It us that ability we wish to capture automatically in an Letter To Sound (LTS) rule system. Trained LTS rules are generally better than hand written ones for complex languages. Writing LTS rules by hand is hard and very time consuming. The appropriateness and difficulty of using LTS rules is very language dependent. As well as the gain of removing entries from lexicon if the LTS rules can predict them correctly. For German and French the lexicon could be reduced up to 90 percent [1] (p.58).

Another important part is prosody. The term is generally used to refer to aspects of sentence's pronunciation which are not described by the sequence of phones derived from the lexicon. Prosody operates on longer linguistic units than phones, and hence is sometimes called the study of suprasegmental phenomena. There are three main phonological aspects to prosody: prominence, structure and tune. Prominence is a broad term used to cover stress and accent. Prosodic structure is described in terms of prosodic phrasing, meaning that an utterance has a prosodic phrase structure in a similar way to it having a syntactic phrase structure. Two utterances with the same prominence and phrasing patterns can still differ prosodically by having different tunes. Tune refers to the intonational melody of the utterance. Intonational tunes can be broken into component parts, the most important of which is the pitch accent. Pitch accents occur on stressed syllables and form a characteristic pattern in the F0 contour.

The three phonological factors interact and are realized by a number of different phonetic or acoustic phenomena. Prominent syllables are generally louder and longer than non-prominent syllables. Prosodic phrase boundaries are often accompany pauses, by lengthening of the syllable just before the boundary, and

sometimes lowering of pitch at the boundary. Intonational tune is manifested in the fundamental frequency (F0) contour.

A major task for a TTS system is to generate appropriate linguistic representation of prosody, and from them generate appropriate acoustic patterns which will be manifested in the output speech waveform. The output of a TTS system with such a prosodic component is a sequence of phones, each of which has a duration and an F0 (pitch) value. This specification is often called the target, as it is this that we want the synthesizer to produce.

Festival contains a number of lexicons for different languages as well as prerecorded voices. There exists automatic processes for building LTS rule systems from lists of entries and their pronunciations.[3].

A phrase break model based on punctuation could be built. There are also standard tricks for intonation and duration prediction. Generally intonation is generated in two steps: prediction of accents and prediction of F0. In the simplest case intonation parameter just set to constant values in the start and at the end of the utterance (130, 110 Hz). More complex modules use so called Classification and Regression trees (CART) - statistical method for predicting data from a set of feature vectors. The tree contains yes/no questions about the features and provides either the probability distribution or a mean and standard deviation. Tones and Break Indices (ToBI) labelling system is implemented in Festival using CART technique. CART could be handwritten or constructed automatically from a set of training data. The default duration model is where all segments are 100 milliseconds. Another simple solution is to use average duration for each phoneme. Some more sophisticated CART duration prediction techniques are also implemented.

In Festival there is a module for post-lexical rules, which is run after accent assignment, but before duration and intonation generation. Post-lexical rules are supposed to take care of sound reduction, insertion and other phenomena, which can not be detected in isolation from the context.

4. Waveform generation

The most natural idea of producing continuous speech from prerecorded small pieces is simple concatenation. In order to make such speech sound smooth we need to use signal processing.

Since the context of the phone affects its pronunciation the elementary unit should be larger than one phone. The longer unit the more natural sound we get. However, there are too many combinations of phones and for the sake of space saving usually diphone model is used. Diphone units normally start half-way through the first phone and half-way through the second. This is because it is known that phones are more stable in the middle than at the edges, so that the middles of most phones of the same phoneme in a diphone are reasonably similar, even if the acoustic patterns start to differ substantially after that. If diphones are concatenated in the middles of phones the discontinuities between adjacent units are often negligible.

The output of diphone synthesizer corresponding to the requested phone sequence still sound unnatural because the prosodi of each phone in the concatenated waveform will be the same as when the diphones were recorded and will not correspond to the pitch and durations requested in the input. The next stage of the synthesis process therefore is to use signal processing techniques to change the

prosody of the concatenated waveform. The standard techniques is time-domain pitch-synchronous overlap and add (TD-PSOLA).

Smoothing diphones by signal processing can produce reasonable quality speech, but the result is not ideal, because of the inevitable distortion and other effects outside the signal processing algorithms. The conclusion is that having a single example of each diphone is not enough. Unit-selection synthesis is an attempt to address this problem by collection several examples of each unit at different pitches and durations and linguistic situations so that the unit is close to the target in the first place and hence the signal processing needs to do less work.

By using a much larger database which contains many examples of each unit, unit-selection synthesis often produces more natural speech than straight diphone synthesis. Some systems then use signal processing to make sure the prosody matches the target, while others simply concatenate the units following the idea that a utterance which only roughly matches the target is better than one that exactly matches it but also has some signal processing distortion.

Within Festival several synthesis both methods are supported. External synthesis methods like MBROLA [5] could also be used.

5. Working with new languages in Festival: the path to follow

In order to add a voice [9] in a new language one need to provide pieces for:

- Phone set
- Token processing rules (numbers etc)
- Prosodic phrasing method
- Word pronunciation (lexicon and/or letter-to-sound rules)
- Intonation (accents and F0 contour)
- Durations
- Waveform synthesizer

So most of the task are related to linguistics analysis, while text analysis and waveform generation remains more or less language independent. One may, in some cases, get away with very simple solutions (e.g. fixed phone durations), or be able to borrow from other voices/languages, but whatever one end up doing, one will need to provide something for each part.

The basic processes to be addressed are:

- construct basic template files
- generate phoneset definition
- generate diphone schema file
- generate prompts
- record speaker
- label nonsense words
- extract pitchmarks and LPC coefficient
- test phone synthesis
- add lexicon/LTS support
- add tokenization
- add prosody (phrasing, durations and intonation)
- test and evaluate voice
- package for distribution

In all cases, these new voices consist of a set of diphones and some scheme code to provide a front end, including text analysis, pronunciation, and prosody prediction. The voices are quite separate from Festival itself, and can be distributed as packages that can be installed against any installation of Festival. The voices do not interfere with any existing, installed voices.

For the most languages and often for new dialects a new phone set is required. It is the basic building block of a voice and most other parts are defined in terms of this set.

Thus, going from the Festival side in order to have synthesis in a new language could be a trying experience with not necessarily acceptable results. System is still under development and hopefully will be improved as well as better documented in the future.

6. Phonetic grammars for GF

Another way to get some speech output in a new language is just to write phonetic grammars in GF, so that instead of letters we have phonemes. Thus, we can skip letter-to-phones step and do not need a lexicon. This approach works for example for Russian, where the relation letter-sound more or less straightforward [11]. There is no such thing like 'ph' or 'th' in English, so it is basically one-two-one correspondence. There are 33 letters in the Russian alphabets to cover all sounds. There are even two letters, that make the preceding consonant soft or hard. Of course vowels in the stressed positions are not exactly the same as in unstressed, they are longer and probably a little louder, but still a certain letter is easily recognizable in every context.

Strong vowels (could be both stressed or unstressed):

Letter	IPA
а	[a]
у	[u]
и	[i]
э	[æ]
о	[o]
е	[e]

Weak vowels corresponding to letters:

Letter	IPA
ё	[jo]
ю	[ju]
я	[ja]

Consonants:

Letter	IPA	Letter	IPA
б	[b]	в	[v]
г	[g]	д	[d]
ж	[ʒ]	з	[z]
к	[k]	л	[l]
м	[m]	н	[n]
п	[p]	р	[r]
с	[s]	т	[t]
ф	[f]	х	[x]
ц	[ʂ]	ч	[t∫]
ш	[∫]	щ	[h]

From the tables one can see that Russian phone set generally could be regarded as a subset to phone sets of the implemented languages like Spanish. Phonetic grammars could be quite acceptable for a limited size domains, which GF is mostly working with. Of course, without building a voice specially for Russian the quality could not be expected to be good enough. Prosody characteristics are quite language dependent as well as token processing rules. Therefore, one could argue if having such pseudo-Russian TTS is reasonable at all. But at least this approach could be implemented quite fast, while building a new voice requires considerably more efforts and competence. Because of technical problems I have not performed any implementation and therefore there was no extensive testing. But one can guess, that the speech would certainly have some strong foreign accent, since phonemes are not exactly matching.

7. Conclusion

Festival was originally developed to be suitable for embedding in other projects that require speech output. Therefore, software engineering was considered very important to the development of Festival. This philosophy and consequence system design and architecture [2] makes Festival relatively easy to use for general TTS purposes. There are a number of languages with prerecorded lexicons supported by Festival. In order to add a new language one could choose to build a new voice, which could be pretty hard, or use the existing voices. First approach is more fundamental but costly, second is much easier, but there is no guarantee of result quality.

References

[1] Alan W. Black, Paul Taylor, Richard Caley (2001). "The Festival speech synthesis system 1.4.2, system documentation". Available from http://www.cstr.ed.ac.uk/projects/festival/manual/festival-1.4.0.ps.gz

[2] Alan W. Black, Paul A Taylor,Richard Caley (1998). "The Architecture of the Festival Speech Synthesis System", in The Third ESCA Workshop in Speech Synthesis, pp. 147-151. Available from http://www.cstr.ed.ac.uk/publications/papers/1998/Taylor_1998_d.ps

[3] Alan W. Black, Kevin Lenzo, Vincent Pagel (1998). "Issues in Building General Letter to Sound Rules", in The Third ESCA Workshop in Speech Synthesis , pp. 77-80 Available from http://www.cstr.ed.ac.uk/publications/papers/1998/Black_1998_a.ps

[4] Thierry Dutoit (1997). "An Introduction to Text-To-Speech Synthesis". ISBN 0-7923-7923-4498-7

[5] MBROLA Work Page. Available from tstc.fpms.ac.be/synthesis/mbrola.html

[6] Aarne Ranta (2001). GF Work Page. Available from www.cs.chalmers.se/~aarne/GF/

[7] Aarne Ranta (1994). "Type-Theoretical Grammar". Oxford University Press.
[8] Aarne Ranta. Grammatical Framework, A Type-Theoretical Grammar Formalism. to be published in J. Functional Programming, 2001
[9] Building Voices in the Festival Speech Synthesis System Processes and issues in building speech synthesis voices Edition 1.2:beta, for Festival Version 1.4.1 10th July 2000 by Alan W Black and Kevin A. Lenzo. Available from http://festvox.org/festvox/festvox_toc.html
[10] Daniel Jurafsky, James H. Martin (2000). "Speech and language processing", Prentice hall.
[11] Olga Krivnova (1999). "Automatic syntethis of Russian speech", V.1. San Francisco. Available from http://isabase.philol.msu.ru/SpeechGroup/publications/congr99.rtf

Chalmers University of Technology
E-mail address: janna@cs.chalmers.se