An enhanced pitch modeling supporting a Greek Text to Speech system

ILIAS SPAIS,   GEORGE BAFAS
Department of Chemical Engineering
National Technical University of Athens
9, Heroon Polytechniou St., Zografou Campus, 15780, Athens
GREECE

and

XENOFON PAPADOPOULOS
Telecommunications Laboratory
National Technical University of Athens
9, Heroon Polytechniou St., Zografou Campus, 15780, Athens
GREECE
xenofon@edu.teiath.gr

Abstract: This paper tries to describe as accurately as possible an enhanced procedure for predicting the appropriate prosodic structure of speech and apply it to a Greek Text to Speech system. The main focus will be a particular linear stylization of the fundamental frequency function F0 contour (pitch), by using the most important syntactic, grammatical and lexical features of the Greek language. Finally, we present an evaluation on the results of plugging the pitch model into the system.

Keywords: speech, synthesis, prosody, pitch, model, rules, vector, TTS engine, attributes

1 Introduction

Speech synthesis is a voice technology that converts raw text input into audible speech. It is a fundamental component of many voice applications and Interactive Voice Response (IVR) systems. Combined with speech recognition, which allows users to provide speech *input* to an application by speaking instead of typing, clicking a mouse or pressing the keys on the phone keypad, speech synthesis is one of the ways you can provide speech *output* for your application. In other words, speech synthesis gives your application its voice.

Speech synthesis is commonly referred to as Text-To-Speech (TTS). In a TTS system, the input text is analyzed, processed, and "understood". Input consists of raw text and, optionally, special tags (known as annotations) that can change the sound of the voice that is ultimately produced. Here’s how it works: First, the TTS system analyzes the word, phrase, or sentence to vocalize. It expands abbreviations, handles contractions and numbers, and disambiguates the semantics of the sentence in order to produce a normalized version of the text to be intoned. Appropriate audio characteristics, such as volume, pitch and speed, are then applied, and the speech output is produced.

Although there are many advantages for using TTS, one of the concerns that has been voiced consistently by users is that TTS voices are not natural enough; that is, the voices do not sound enough like human voices.

In the field of speech synthesis prosodic structure of speech is a hot topic. TTS still suffers to some extend from unnaturalness. Although in the past few years great progress has been made in this field, the dislocation in prosodic hierarchy seems to cause a lot of specific problems. Furthermore, lacking a firm understanding of the prosodic structure hinders the improvement of the accuracy in speech recognition.

In synthetic speech, overall loudness, emphasis, and pitch changes are the basic features of prosody in speech processing. Many of the differences between human and synthetic speech are due to the fact that these features are extremely difficult to be recreated. In human speech, prosody reveals a great deal of information about the speaker’s personality, mood, what he wants to emphasize, and characteristics of the topic being discussed. In synthetic speech, prosody and coarticulation are by-products of how individual sounds are sequenced, and are less likely to have the same loudness, pitch, rate, or emphasis as when the same words are spoken by a human. Consequently, it is true to say that prosody is the area in which the most progress needs to be made before such technology can be used as an acceptable replacement for human speech, in both speech synthesis and speech recognition fields.

This paper describes a statistical pitch model for a synthetic Greek Text-To-Speech system. The model has the power to provide smoother, more natural-sounding voices based on improved expressiveness and intonation. The improved naturalness can increase both the understandability of the intoned speech and the level of user's acceptance and satisfaction.

The main task in this model is to segment syllable sequence into proper units and then organize them into correct pitch layers based on text analysis. These units are called vectors and each vector's attribute takes a value by applying to the unit syntactic grammatical and lexical rules. These rules are the result of a research on the special characteristics of Greek language. After generating all the vectors, the procedure interrelates each vector with a specific pitch value depending on vector's position into F0 contour. In this way we create a phonetic-vector training database. Whenever there must be a text to speech extraction, the model selects the most similar to the input vector from the donor's database. Finally, by using its corresponding pitch value we generate the pitch contour anchor points.

The paper has the following structure: Section 2 presents the basics of a Greek Text To Speech System, while section 3 describes with every possible detail the pitch model which improves system's output. Section 4, concludes the paper and list ideas for extending this work.

2 General Description of TTS

TTS process is performed by a software component known as the Text-To-Speech engine. The primary function of the text-to-speech engine is to process text input and translate it into spoken output. To do this, the TTS engine employs two major components: a) the Text Preprocessor and b) the Synthesizer. The Text Preprocessor expands abbreviations and numerals, and disambiguates the semantics of the sentence in order to provide a normalized form of the input text which is subsequently passed to the Synthesizer. The Synthesizer then uses suitable algorithms to convert the text into speech.

Our engine's synthesizer is based on a phoneme-selection concatenative algorithm, which attempts to select the suitable segments of speech from a repository of recorded voice and join them to produce new spoken text. The repository has been created by recording 1580 Greek sentences, aligning the text with phonemes using a language model of the Greek language, and storing the speech segments and the corresponding phonemes in a database.

In order to synthesize a new sentence, the synthesizer converts the text to a sequence of phonemes and attempts to select the appropriate voiced form of each phoneme from the segments pool. The selection is performed by using purely statistical methods, by examining the position of each phoneme in each word and its relation with the nearby phonemes. Finally, the synthesizer concatenates the selected segments, thus producing the necessary speech.

In a slight variation of this algorithm which is not based exclusively on phonemes, longer segments of recorded speech are stored during the training and selected during the synthesis. When these recorded speech units are entire words, phrases or even sentences, the output can be very natural, human-sounding speech.

3 Pitch Modeling

The fundamental frequency F0 is the feature of prosody that our model tries to predict in order to make the TTS acceptable. Due to the fact that it is extremely difficult to create such a characteristic from scratch, the F0 extraction algorithm is based on creating an appropriate pitch database by using a given corpus of spoken Greek sentences. The database consists of vectors and each vector's attribute comes out by applying to each speech segment unit (phoneme) specific syntactic grammatical and lexical rules of the Greek language. Essentially, the module has the power to obtain the variability of pitch in a Greek spoken sentence by modeling a given F0 contour.

3.1 Analyzing Corpus

Our speech corpus consists of 1580 Greek spoken sentences. In order to eliminate noise the speech signal was recorded in a studio using 22 KHz sampling rate. In parallel with the microphone we used laryngograph**,** a tool which can provide us the basic features of prosody including F0 contour. In this way for each one of the 1580 sentences that we recorded, we created two files: a) the .wav file that contains the speech signal and b) the .lar file that contains all the useful prosody features of the sentence. By elaborating these two with specific filters and tools and by taking advantage of the results of processing with appropriate acoustic and language models, we managed to correlate units of speech waveforms with a concrete phoneme. In this way, we accomplished to segment each sentence to phonemes embodied with prosody features (duration, start_time, pitch etc.). In essence, we transformed each one of the 1580 sentences to files (.occ) that can describe the variability of F0 contour for each sentence.

Table 1 shows the results of analyzing the word "arrival" («άφιξη»). Each phoneme of the word is represented by three leaves and each one of them has prosody features including pitch values. For example the phoneme "_F." which represents the Greek letter «φ», has the leaves "_F.1" (pitch – F0 value = 111.4), "_F.2" (pitch – F0 value = 115.7) and "_F.3" (pitch – F0 value = 119)*.

occ_start_time	leaf_name	start_pitch	end_pitch	dur_leaf_name
1..39	A_A1(386)	95.8	97	A_A1(111)D
1.43	A_A2(463)	97	106	A_A2(111)D
1.47	A_A3(507)	106	111.4	A_A3(111)D
1.50	_F.1(3133)	111.4	115.7	_F.1(1153)D
1.53	_F.2(3161)	115.7	119	_F.2(1153)D
1.57	_F.3(3173)	119	120.5	_F.3(1153)D
1.59	I_S1(1827)	120.5	122.4	I_S1(726)D
1.60	I_S2(2015)	122.4	117.3	I_S2(726)D
1.61	I_S3(2098)	117.3	121.4	I_S3(726)D
1.63	_K.1(3303)	121.4	120.5	_K.1(1238)D
1.64	_K.2(3348)	120.5	120	_K.2(1238)D
1.65	_K.3(3395)	120	119	_K.3(1238)D
1.67	_S.1(4392)	119	118	_S.1(1754)D
1.68	_S.2(4525)	118	116.3	_S.2(1754)D
1.70	_S.3(4686)	116.3	116	_S.3(1754)D
1.71	I_S1(1720)	116	115.3	I_S1(731)D
1.72	I_S2(1853)	115.3	114.05	I_S2(731)D
1.74	I_S3(2099)	114.05	112	I_S3(731)D

Table 1. Prosody features for the word arrival

Finally, the initial step of the procedure ends by taking into account the average value of pitch for each phoneme (Phoneme _F. has average pitch = 115.37).

3.2 Rules specifications – Training database

The next step is to transform phonemes to vectors with specific attributes. The whole step is based on applying concrete rules of Greek language to each one of the phonemes in order to generate vectors with a respectively value of pitch.

3.2.1 Rules architecture for Greek language

In the framework of rules architecture, we aim at computing attributes for each vector in such a way that it can be used to provide pitch values to the system. To achieve this goal we had to answer two main questions: what kind of attributes-rules we must generate and how many of them.

It is beyond any doubt that the quality of the attributes must be based on syntactic, grammatical and lexical hints of the Greek language. Several factors support such a point of view. First of all, as many researchers have found, prosodic hierarchy is not always consistent with syntactic hierarchy in a language. At which point the latter can be true depends on the language. Furthermore, we cannot expect to determine prosodic words directly according to lexical words. Last but not least, is the fact that the Greek language has many grammatical features that are quite different from other languages, such as its flexibility and poly-synthesis in word formation. Consequently, the attributes must be combinations of syntactic, grammatical and lexical characteristics. As far as the number of rules is concerned, neither too many nor too few of them should be used; having "too many" rules will lead to the creation of unique vectors that probably may not be used from the Text to Speech system. On the other hand by having "too few" will certainly lead to an inadequate presentation of F0 contour. Determining the optimum number of rules is an issue that should also be addressed.

By examining in details the performance of the TTS in several experiments that we conducted, we managed to create 15 attributes for each vector. We can group these attributes in five categories:

i) Attributes that deal with the quality of the phoneme (constant, vowel etc). There are three rules in this group: the first one deals with the present phoneme, the second deals with the phoneme before the present one and the third with the phoneme after the present one. In other words, we deal with a frame of 3 phonemes.

ii) In the second group the rules present information about the quality of the word. Due to the polymorphism of Greek language it is not possible to detect the syntactic or grammatical role of each word unambiguously, but the algorithm can give an acceptable estimation for the majority of the corpus words. This group consists of three rules just like the previous one. The difference here is that these particular rules analyze words and not phonemes.

iii) One of the basic grammatical features of the Greek language is intonation. In each word there is at least one letter (vowel) with tone or other contextual feature annotations. The two rules in this group deal with this tone: the first one reports which phoneme has the

* In order the results of this word analysis to be more legibly, we present approximately values and not accurately ones.

annotation and the second one reports if there is more than one vowel in the word.
iv) Attributes that deal with the spelling of the word. Trying to be as accurate as possible we generated virtual spelling guidelines based on the Greek grammar. In this way the model can overcome difficulties that may occur by using strictly real ones. This group consists of five rules: number of syllables for each word, at which syllable the phoneme belongs to, distance (in syllables) of the phoneme from the end of the word, distance from the syllable which has the intonation and distance from silence or break.
v) One of the basic syntactic features of the Greek language is that in a sentence pitch changes remarkably at points where there are some key words or specific syntactic annotations (e.g.“,”). After carrying out a research we accomplished to record these annotations and some of these words. The attributes in this category provide to the vector the distance of the phoneme from such a word or annotation (in this group the algorithm counts words and not syllables). The last attribute exposes the name of the phoneme.

Finally, we must add that these rules are also applied to phonemes that present pause in a sentence.

3.2.2 Generate training database

As we reported before the rules represent the attributes of each vector. By sequence, each vector represents a phoneme with a respective value of pitch. Consequently, by applying the rules to each phoneme we accomplished to generate vectors with concrete attributes and with a specific value of pitch. These attributes supply vectors with suitable characteristics in order to discriminate them. For example, the phoneme _F. with pitch 115.37 has been transformed to vector [_F.,22,1,1,0,1,1,6,3,1,1,1,-1,2,-800] with the same pitch. The pitch database consists of all the generated vectors.

3.3 Apply pitch model to TTS

The main task of the implemented Greek Text to Speech system is to produce speech by synthesizing units of human speech. The procedure starts by analyzing the word, phrase, or sentence to vocalize. It expands abbreviations, handles contractions and numbers, and then it disambiguates the semantics of the sentence in order to create a normalize form of the text. Subsequently, once the sentence, word, or phrase has been analyzed, the system can then determine exactly how to synthesize these units (phonemes) into speech by using the concatenative method we introduced before.

In order to embody speech with the appropriate audio characteristics (such as pitch), the system activates the model. Initially, the model uses the input sequence of phonemes and applies the rules to each one of them. In this way it creates an input vector S(i) (i stands for the number of attributes) for the entire phoneme. The next step, which is the most crucial, is to select prosody vector P(i) from the training database which is most similar to the input vector S(i). In essence, the model generates the F0 contour anchor points by assigning to each vector S(i) a pitch value.
However, there are two exceptions which generate difficulties, although they do not appear so often: a) if no clause pattern in the database is sufficiently close by the matching, every individual phoneme in the input string will be assigned a high quality pitch based on the neighboring values and b) if there is more than one matching vector P(i) the model assigns to the input vector the average value. Finally, we must underline that the model may perform even better if we enrich it with stochastic methods in order to overcome these two exceptions.

4 Conclusions and Perspectives

The procedure presented here gives access to the hidden structure of intonation. By applying the model to the TTS, the system can actually capture significant prosodic variations with a rather few number of prototypical movements, generates faithful and varied prosodic contours. We demonstrated the architecture of the rules which are the basic feature of the training database of the model.

After conducting informal listening tests, the results that we have recorded by applying the model to the system are encouraging. However, they could be probably improved if we will use more information regarding Greek language, in order to generate more reliable rules. For instance, we must improve the rules on the quality of the word. As we said before due to the polymorphism of Greek language it is very difficult to find out if one word is a verb or adjective or adverb. At this point the model reports an evaluation and not certainty. Additionally we must improve the spelling of each word. We must overcome all the difficulties and generate real and not virtual spelling. Furthermore, the informal tests we carried out has proved that we should focus our efforts on generating rules which will come out as a result of investigating and processing deeply Greek speech from syntactic and grammatical point of view. After stabilizing at a solution for the above problems, it will be feasible to conduct formal experiments by using the appropriate methods, in order to estimate the results of applying the model to the system and compare it with existing methods.

Based on Text To Speech theory and Speech Recognition, several Natural Language Understanding (NLU) systems can be implemented. To be more specific, TTS system is the core feature of conversational applications (e.g. telephony application). The overall objective of such an application is to give the user the opportunity to have a conversation with the machine and interact with it, resembling talking to a human operator. The role of TTS is to translate the responses of NLU application to audio prompts. Finally, by taking the above hints into account, we can report that an enhanced TTS system supported by a pitch model like the one that we described is a necessity for that kind of applications, due to the fact that it can produce natural voice just like a human voice.

References:

[1] R. E. Donovan, E. M. Eide (1998) “The IBM Trainable Speech Synthesis System”.

[2] R. E. Donovan, A. Ittycheriah (2002) “Current Status of the IBM Trainable Speech Synthesis System”.

[3] Paul C. Bagshaw (1998) “Unsupervised Training of Phone Duration and Energy Models for Text-To-Speech Synthesis”.

[4] Merle Horne (2000) Prosody, Theory and Experiment: Studies Presented to “Geosta Bruce” (Text, Speech, and information Technology).

[5] Stavroula-Evita F. Fotinea, Michael A. Vlaxakis and George V. Carayannis “Modeling arbitrarily long sentence-spanning F0 contours by parametric concatenation of word-spanning patterns”, Rhodes, Greece: ESCA Eurospeech97, Sep 1997, vol 2, pp.315-318.

[6] Stavroula-Evita F. Fotinea, “Sentence-level Prosodic Modeling of the Greek language with Applications to Text-To-Speech synthesis”, PhD Thesis (in Greek), National Technical University of Athens, University Press, 1999.

[7] Thierry Dutoit ‘High – quality Text-to-speech synthesis: an overview’.