A Complemented Greek Text to Speech System

XENOFON PAPADOPOULOS
National School Network
TEI of Athens
Ag.Spuridonos & Milou 1, Aigaleo, Athens
GREECE

and

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

Abstract: This paper tries to give a comprehensive insight of a complemented Greek Text to Speech system by highlighting its basic Digital Signal Processing (DSP) and Natural Language Processing (NLP) modules. The main focus will be the development of such a system by taking into account syntactic, grammatical, phonological and lexical knowledge of Greek language. The ultimate goal is to boost up academic research on speech synthesis, particularly on graphemes to phonemes transcription and on prosody generation, known as two of the most important challenges in Text to Speech Synthesis for the years to come. Furthermore, we introduce a conversational application which uses the system and present an evaluation on the results of this application.

Keywords: speech, synthesis, text preprocessing, grapheme, phoneme, transcription, pitch, model, TTS engine

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 keys on the phone keypad, speech synthesis is one of the ways to provide speech output for an application. In other words, speech synthesis gives your application its voice [2].

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 [1]. 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. Consequently, there is a fundamental difference between the system we are about to discuss here and any other talking machine (like a pre-recorded speech playback engine) in the sense that we are interested in the automatic production of new sentences. It is thus more suitable to define Text to Speech as the automatic production of speech, through a grapheme to phoneme transcription of the sentence to utter.

At first sight this task does not look hard to perform. After all, is not the human being potentially able to correctly pronounce an unknown sentence, even from his childhood? We all have a deep knowledge of the reading rules of mother tongue. However, it would be a bold claim indeed to say that it is only a short step before the computer is likely to equal the human being in that respect.

Due to the emergence of new technologies (e.g. Natural Language Processing techniques), it is now a necessity for TTS systems to be enhanced in order to provide high quality synthesis. This paper describes such a complemented TTS system, appropriate for the

Greek language and supported by several components in order to be used in such technologies.

In the field of speech synthesis graphemes to phonemes transcription and prosodic structure of speech are the basic procedures that must be optimum in order for the TTS system to be reliable and capable of producing natural voice just like a human voice. Consequently, the main focus of this article will be the description of specific modules that produce the correct transcription of words into phonemes and calculate the prosodic contour of the sentence to be synthesized.

The paper has the following structure: Section 2 presents a functional overview of a Greek Text To Speech System, while section 3 describes with every possible detail the basic components of the system. Section 4, evaluates system's performance and section 5 presents potential applications. Finally, section 6 concludes the paper and list ideas for extending this work.

2 Functional Overview of a Greek TTS

TTS process comprises a Natural Language Processing module (NLP), capable of producing a phonetic transcription of the text read, together with the desired intonation and rhythm (prosody), and a Digital Signal Processing module (DSP), which transforms the symbolic information it receives into speech.

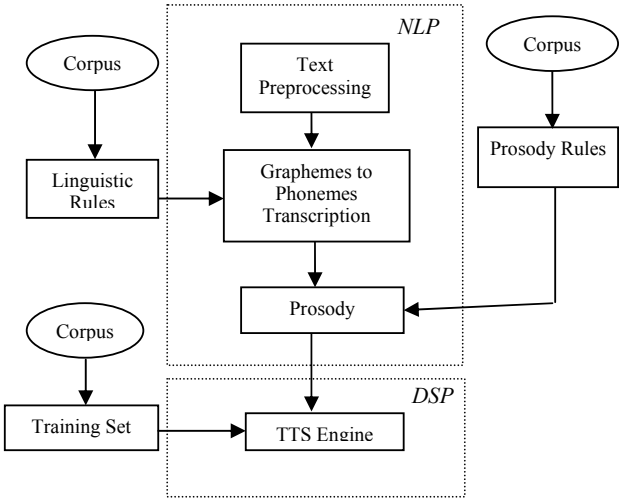

Fig.1.Overal structure of a TTS system for Greek language

Fig.1 shows the structure of our system which is in line with the functional organization of a general Text to Speech Synthesizer. NLP module consists of: a) Text preprocessing, b) graphemes to phonemes transcription and c) natural prosody. Text Preprocessor expands abbreviations and numerals, and disambiguates the semantics of the sentence in order to provide a normalized form of the input text. Graphemes to phonemes transcription require linguistic rules in order to generate a sequence of phonemes from the text and prosody model produces pitch and duration for each one of these phonemes.

These enhanced phonemes are subsequently passed to the TTS engine, which is the basic component of the DSP module. Finally, the engine uses a phoneme-selection concatenative algorithm, which attempts to select the suitable segments of speech from a repository of recorded voice and join them in order to produce new spoken text.

3 Analyzing the Basic Components

3.1 Text preprocessing

A generic speech synthesizer has no control over the type, the content or the quality of the text it should synthesize. The text may contain spelling mistakes, lack punctuation or include foreign words. It may also contain numerics, abbreviations or acronyms that must be expanded to a longer form before their utterance. Since a synthesizer should generate speech for any given input, a process of preprocessing and filtering the text prior to the actual synthesis is essential. This normalization of the text is language dependent; although some problems are addressed in virtually the same way across different languages, a special approach is often required depending on the grammatical and syntactical structure of a specific language.

We should note that in some case the normalization of a given text may result into several different, yet equivalent, outputs. For example, a telephone number is often uttered in several different ways, depending on the speaker's preferences. In these cases, it suffices to select any valid normalized output.

The fundamental problem in handling Greek text occurs during the expansion of the abbreviated forms of conjugated words. Ordinal numeric is a typical example of this problem: the number *21* is expanded into a different word in the phrases shown on Table 1.

Greek phrase	**Meaning**	**Expanded form of '21'**
Θα έρθω σε 21 ημέρες	I will come in 21 days	εικοσιμία
Η επέτειος των 21 χρόνων	The 21-years anniversary	εικοσιενός

Table1. Example of expanding number 21 for two different phrases.

In the second case of the example, it is relatively easy to determine the gender of the number, since the leading article '*των*' denotes male (or neutral, which has

the same expanded form) gender, in plural. It is not so in the first case, however. A complete solution of this problem requires the development of a grammatical analyzer that will tag each word in a sentence with its grammatical attributes. Lacking a complete analyzer, a simple tokenizer and a parser can be used to guess the grammatical attributes of each word with fairly high accuracy.

The possible conjugation of the Greek words turns the expansion of abbreviations into a complicated problem. Although it is simple to detect an abbreviation in any given text, determining the original word often requires the grammatical analysis of the context. For example, *κ.* is used for both Mr. and Mrs. To complicate matters even more, the expanded form of this abbreviation changes depending on the case. Assuming that *κ.* stands for the male world 'mister', if its context is *of mister* it would expand to *του κυρίου,* while in a *to mister* context it would expand to *τον κύριο*. As we already mentioned, a method of determining the grammatical attributes of each word in the sentence is required. Once the attributes are known, the normalization may occur through the use of a dictionary of abbreviated words.

The expansion of acronyms is addressed in a way almost identical to that of abbreviations. There is, however, one exception: in some cases, an acronym should not be expanded at all; the synthesizer should instead utter its abbreviated form. For example, it is perfectly legit to expect a synthesizer to utter *the USA* as *the U - S - A,* instead of *the United States of America.* In this case, no grammatical analysis is required, since acronyms are never conjugated in Greek.

Apart from the typical problem of the grammatical attributes of a word, the normalization of numerics presents yet another difficulty: the detection of the type of numeral. For example, *16:45* denotes the time, while *(+30) 210 7776273* is a telephone number and *(3 + 10)* is a simple mathematical expression. Apparently, different normalization rules apply to each case. A lexical analyzer can be used to determine the correct type of a numeral, with high accuracy. For the more complicated cases, however, more advanced means of text analysis are required. As an example, even if a lexical analyzer correctly classifies *10 / 2* as a date instead of some mathematical expression (division), it would need additional locale information in order to determine whether it stands for the 2nd of October or the 10th of February.

In our system, we used a set of Flex rules to identify the grammatical attributes of each token. In order to evaluate its efficiency, we randomly selected 2,000 sentences taken from the archive of the Proceedings of the Greek Parliament, rich in abbreviated forms. The results are presented in Table 2.

	Identified	**Mistaken**	**Tagged**
Abbreviations	91.30%	0.25%	85.73%
Acronyms	98.10%	0.01%	88.49%
Numerics	100.00%	7.62%	91.23%

Table 2. Results of identifying grammatical attributes.

Identified stands for the percentage of abbreviated forms actually identified as such by the tokenizer. *Mistaken* has different meaning depending on the type of the abbreviated form. In the case of abbreviations and acronyms, it is the percentage of words identified as such without being so. In the case of numerics, it is the percentage of identified forms that were thought to be of the wrong type. *Tagged* stands for the percentage of correctly identified forms that were assigned the right grammatical attributes.

3.2 Graphemes to phonemes

The conversion of written text into a sequence of phonemes is a fundamental step in the text-to-speech process. The implementation of such a conversion is language dependent, its approach depending on the glossological structure of each language. It is thus necessary to examine that very structure and attempt to detect the rules (explicit or implicit) that govern the graphemes to phonemes transcription for a particular language.

A study of the Greek language shows that the phonetization of a word is not a purely deterministic process; although rules to determine the conversion of each phone into the corresponding phoneme do exist, there is still some unavoidable ambiguity in the process. This can be explained by the historical evolution of the Greek language, especially during the last 40 years: as punctuation signs that used to denote the pronunciation of a word have gradually disappeared from written text, the utterance of a word can no longer be directly derived from its written form. Disregarding this fact, one can nevertheless study the glossological structure of the language to extract a set of rules that can be used to phonetize Greek text with sufficient accuracy for a preliminary speech synthesis system. The phonetical dictionary created by this process can be then manually refined, to increase the quality of the transcription and bring it to par with high-end speech synthesizers. In the following section we will describe these rules and provide some examples of phonetization of Greek text. Throughout our presentation, we use the IPA standard for the phonetic representation of text.

Each phone in Greek consists of either a single letter, or two adjacent letters (diphones). As all transcription rules apply to phones instead of letters, the first step is splitting a word (i.e. a sequence of letters) into a sequence of phones. This is a purely deterministic process, easily represented through a one token look-ahead automaton.

During the transcription of Greek text, each phone in a word is either converted to a phoneme, or gets muted, disappearing completely in the utterance of the word. In order to generalize the process, we assume that each muted phone is converted to the *null* phoneme. With this assumption, we can say that each phone is always converted to a phoneme.

Our studies have revealed that the transcription of each phone depends (at most) on the two adjacent phones on the left and the two adjacent phones on the right of that phone. It thus suffices to examine a frame of five subsequent phones to convert each phone - the central phone of the frame - into the corresponding phoneme; all the phones outside this frame have no effect on the phonetization. We can then determine a set of rules of the form:

$$L\ \{ f_{-2}, f_{-1}, f_{+1}, f_{+2} \} \rightarrow P$$

Each one of these rules will convert the L phone to the P phoneme, if the phone is surrounded by the $f_{-2}, f_{-1}, f_{+1}, f_{+2}$ phones.

As we already mentioned, there is an innate ambiguity in the Greek language, making it impossible to correspond a single phoneme to each phones frame. Taking this ambiguity into account, our rules will actually be of the form:

$$L\ \{ f_{-2}, f_{-1}, f_{+1}, f_{+2} \} \rightarrow P_1 \mid P_2 \ldots \mid P_n$$

Examining the transcription rules, we realized that phones belonging in a certain class (e.g. vowels, consonants, and certain diphones) have exactly the same effect with each other in several rules. We could then reduce the number of our rules, by using the class of each phone instead of the phone itself, thus giving our rules the following form:

$$L\ \{C(f_{-2}), C(f_{-1}), C(f_{+1}), C(f_{+2}) \} \rightarrow P_1 \mid P_2 \ldots \mid P_n$$

Obviously, a class may contain only one phone, in which case C(f) ≡ f. In addition to this trivial case, we defined the classes of phones that are demonstrated at Table 3. In Table 4, we present an extract of the complete set of transcription rules we generated.

For the transcription of a Greek word, we generate a frame of five phones for each phone of the word and search for a matching rule on the table of transcription rules. This search is sequential, and stops when we find a rule that matches the phones of the frame. This means that the ordering of the rules is very important. An obvious side effect is that each phone has an *L {any, any, any, any} → P* rule assigned to it as the last possible match for the transcription of *L*.

any	It contains every phone of the Greek language. It is used when an adjacent F phone does not affect the utterance of the L phone of the rule.
vowel	It contains all the stressed and unstressed vowels of the Greek language.
consonant	It contains all the consonants of the Greek language.
ext_vowel	It contains all the stressed and unstressed vowels of the Greek language, as well as certain diphones: *ευ*, *εύ*, *αυ* and *αύ*.
start	This is a special class that contains no phones, but denotes the beginning of a word. It is used when the utterance of the L phone depends on its distance from the beginning of the word.

Table 3. Definition of phone's classes.

#	L	C(f$_{-2}$)	C(f$_{-1}$)	C(f$_{+1}$)	C(f$_{+2}$)	P
18	ή	any	any	any	any	'i
23	μ	any	any	φ,β	any	ɱ
24	μ	any	any	any	any	m
29	ο	any	any	any	any	o
33	σ	any	any	β,γ,δ,ρ,μ	any	z
34	σ	any	any	any	any	s
63	μπ	any	any	τ	any	m
64	μπ	-	start	any	any	b
65	μπ	any	any	any	any	mb | b

Table 4. Transcription rules.

As an example, we describe the process for the transcription of the word 'σμήνος' (swarm):

- First, we convert the word into a sequence of phones: σ – μ – ή – ν – ο – σ.
- For each phone, we generate a frame of that phone and its adjacent ones: (-,-,**σ**,μ,ή), (-,σ,**μ**,ή,ν), (σ,μ,**ή**,ν,ο), (μ,ή,**ν**,ο,σ), (ή,ν,**ο**,σ,-) and (ν,ο,**σ**,-,-).
- We search the transcription table for a matching rule for each frame. Matching rules are # 33, 24, 18, 26, 29 and 34 respectively.

- We concatenate the phonemes derived by these rules. The result, ***zm'inos***, is the phonetized form of the word.

In the general case, a word consisting of N phones ($L_1L_2...L_N$) will be converted to a sequence of possible phonemes $\{P_{11}|P_{21} ... |P_{m1}\}$ $\{P_{12}|P_{22} ... |P_{m2}\}$ ... $\{P_{1N}|P_{2N} ... |P_{mN}\}$. This is the effect of ambiguity, and results to a total of $(m_1 \ x \ m_2 \ x \ ... \ x \ m_N)$ possible phonetizations of the word. Although the magnitude of this is massive at first glance, in practice most rules are unambiguous, meaning that $m_i = 1$, while $m_i < 3$ in any case. This means that our transcription method results in transcription of adequate quality and precision.

3.3 Pitch modeling

In the field of speech synthesis prosodic structure of speech is a hot topic. TTS still suffers to some extend from unnaturalness. Although great progress has been made in this field in the past few years, 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.

The fundamental frequency F0 (pitch) is the feature of prosody that our model predicts in order to make the TTS acceptable. The main task in this procedure 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.

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. These attributes display if the phoneme represents a constant letter or a vowel (three rules in this group).

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 (three rules in this group).

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.

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).

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.

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.

3.4 Text to Speech synthetic engine

Text to Speech synthetic engine 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.

4 Evaluating System's Performance

In order to examine the consistency and quality of our observations, we carried out a series of experiments on

the TTS system. A group of native Greek speaking listeners was used in order to rate the quality of the synthetic speech. They were given a small corpus of 9 affirmative sentences, which was utterance by our system. The listeners had to provide to provide a score for each sound sample, based on Mean Opinion Score (MOS). The MOS is the arithmetic mean of all the individual scores, and can range from 1 (worst) to 5 (best).

The mean opinion scores are shown in Fig 2 where one can note that the listeners rated system's performance acceptably. The polymorphism of Greek language affected in the worst way sentence 8, while sentence 5 was utterance almost naturally.

5 Potential Applications of the TTS

Potential applications of high quality TTS Systems like the one that we described are numerous [8]. Here are some examples:

- Telecommunications services. TTS systems make it possible for the users to access textual information over the telephone.
- Aid to handicapped persons. Blind people are widely benefited from such systems when coupled with OCR. Essentially, this cooperation gives them access to written information
- Language education. In this case the system can be coupled with a computer aided learning system and provide a helpful tool to learn a new language.

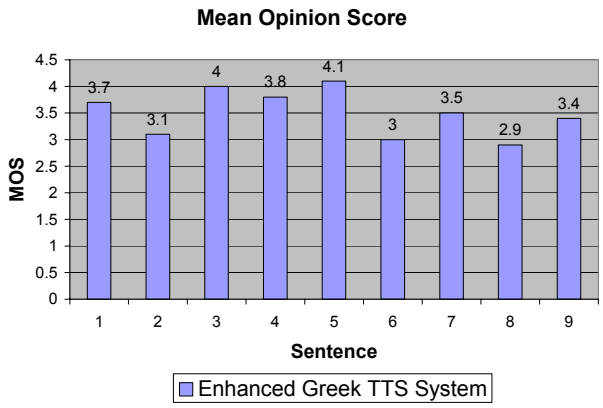

Fig.2. Mean Opinion Score

As far as telecommunication services, 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.

To sum up, by taking the above hints into account, we can report that an enhanced TTS system is a necessity for that kind of applications, due to the fact that it can produce natural voice just like a human voice.

6 Conclusions and Perspectives

This article gives access to the hidden structure of a TTS system. We demonstrated the basic components and modules of such a system and we introduced solutions for the hot topics of Synthesis like the phonetic transcription and the generation of prosody.

After conducting informal listening tests, the results that we have recorded are encouraging. However, they could be probably improved if we will use more information regarding Greek language, in order to generate more reliable models. It is beyond any doubt that the quality of the system depends on syntactic, grammatical and lexical hints of the Greek language and further research must be carried out.

References:

[1] Ilias Spais, George Bafas and Xenofon Papadopoulos "An enhanced pitch modeling supporting a Greek Text to Speech system", Tenerife, Canary Islands, Issue 10, Vol 3, pp.2168, December 2004.

[2] R. E. Donovan, E. M. Eide (1998) "The IBM Trainable Speech Synthesis System".

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

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

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

[6] 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.

[7] 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.

[8] Thierry Dutoit 'High – quality Text-to-speech synthesis: an overview'.