DIXI – A Generic Text-to-Speech System for European Portuguese

Sérgio Paulo, Luís C. Oliveira, Carlos Mendes, Luís Figueira, Renato Cassaca, Céu Viana[1] and Helena Moniz[1,2]

L^2F INESC-ID/IST, [1]CLUL/FLUL, [2]L2F INESC-ID
Lisbon, Portugal
{spaulo,lco,cmdm,luisf,rmfc,mcv,helenam}@l2f.inesc-id.pt

Abstract. This paper describes a new generic text-to-speech synthesis system, developed in the scope of the Tecnovoz Project. Although it was primarily targeted at speech synthesis in European Portuguese, its modular architecture and flexible components allows its use for different languages. We also provide a survey on the development of the language resources needed by the TTS.

1 Introduction

This paper describes a new generic text-to-speech (TTS) synthesis system, developed in the scope of the Tecnovoz Project. Although it was primarily targeted at speech synthesis in European Portuguese (EP), its modular architecture and flexible components allows its use for different languages. Moreover, the same synthesis framework can be used either for limited domain or generic speech synthesis applications. The system's operation mode is defined by the currently selected voice, enabling the user to switch from a limited domain to a general purpose voice, and vice-versa, with a single engine. Dixi currently runs on *Windows* and *Linux*. The synthesis engine can be accessed, in both operating systems, by means of an *API* provided by a set of *Dynamic Linked Libraries* and *Shared Objects*, respectively.

Given the success enjoyed by the Festival Speech Synthesis System [3] and the flexibility of its internal representation formalism, the heterogeneous relation graphs [14], the Dixi's internal utterance representation follows approximately the same scheme. However, the Festival system implementation has a large number of drawbacks that led us to the implementation of a new system architecture. One of the limitations of the Festival system is its inability to use multi-threading, and thus incapable to profit from the multi-processing capabilities of nowadays machines. Being multi-thread safe is a key feature of the new system. The system architecture is based on a pipeline of components, interconnected by means of intermediate buffers, as depicted in Fig 1. Every component runs independently from all others, loads the to-be-processed utterances from its input buffer and, subsequently, dumps them into its output buffer. Buffers, as the name suggests, are used to store the utterances already processed by the previous component while the following one is still processing earlier submitted data.

A. Teixeira et al. (Eds.): PROPOR 2008, LNAI 5190, pp. 91–100, 2008.

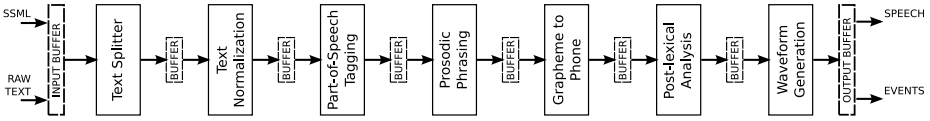

Fig. 1. Overview of the system architecture, where *SSML* stands for Speech Synthesis Markup Language

The capability of the system to use multi-processing and split large utterances into smaller ones, as will be explained later on in this paper, allows the streaming synthesis problem to be addressed more efficiently than in other well known synthesis systems [3,1]. Dixi comprises six components: text pre-processing, part-of-speech tagging, prosodic phrasing, grapheme-to-phone conversion, phonological analysis and waveform generation, as depicted in Fig 1.

1.1 Tecnovoz Project

The Tecnovoz project is a join effort to disseminate the use of spoken language technologies in different domains of application. The project consortium includes 4 research centers and 9 companies specialized in a wide range of areas like banking, health systems, fleet management, access control, media, alternative and augmentative communication, computer desktop applications, etc. To meet the goals of the project a set of 13 demonstrators are being developed based on 9 technology modules. Two of these modules are related with speech output: one module for limited domain speech synthesis and another for synthesis with unrestricted input. The first module will be used, for example, in banking applications were almost natural quality can be achieved by a proper design of the output sentences. An example of an application with unrestricted vocabulary is the oral feedback for a dictation machine.

We decided to adopt a single system to handle both requirements. The domain adaptation is performed at the level of the speech inventory used for each application. The inventory, usually called the system "voice", can have a wide or narrow coverage of the language. By using an inventory with very large number of carefully selected samples of a restricted domain, a very high quality can be achieved for sentences in that domain. A more general purpose system can use an inventory with a wider coverage but with fewer examples for each domain.

1.2 System Flexibility

The TTS users can adapt the system operation to their own needs by themselves. Accordingly, the users can create an addenda to the pronunciation lexicon, in order that words are rendered as desired. Moreover, specific normalization parameters[1] can be specified by the user, so that some particular text tokens are normalized according to the user-specific needs.

[1] Such as language, regional settings and text domain.

The speech signal can be produced by two distinct approaches. Parametric synthesis, using Hidden Markov Models based synthesis [15], or concatenative synthesis, using variable-length units [2].

1.3 Data-Driven Approaches

In order to accelerate the system's adaptation to new languages and domains, the language- and domain-specific knowledge sources were kept apart from the system's implementation. Also, machine learning techniques were used to train models for some components responsible for the linguistic analysis of the input text. The models – frequently encoded in the form of *Classification and Regression Trees* (CART) [5] – are then loaded the same way no matter what domain or language the system is dealing with.

1.4 Paper Organization

Although being a multi-lingual synthesis system, in this paper we will focus on the specific needs for speech synthesis in European Portuguese, namely, corpora and linguistic analysis. The paper is organized as follows. In section 2, we describe the corpora building procedures for limited domain and generic synthesis applications. Section 3 is reserved for describing the training of speaker-specific prosodic phrasing models, as well as the building of grapheme-to-phone (G2P) conversion models. In section 4, we present a detailed description of the system architecture. Conclusions and future work are presented in section 5.

2 Corpora Building

The quality of the synthetic speech produced by a corpus-based synthesizer depends, to a large extent, upon the suitability of the speech inventory to represent the variability of the language within the target application domain. While it is quite easy to design a set of limited domain sentences comprising units such as words in appropriate prosodic contexts, dealing with an unrestricted text task calls for another corpus design approach. In such a case, it is impossible to list all the target domain words, as unknown words can always appear in the TTS input. Hence, the representation of the language must be addressed by means of a finite set of linguistically motivated units (e.g. phonemes, diphones or syllables). Besides, several corpus design strategies can be chosen, as prompts can be manually designed or automatically selected from a a huge candidate sentence set. The design of limited domain and open domain speech inventories are described separately in this section, as they consist in problems of distinct nature.

2.1 Corpus Design

We followed a mixed approach for designing both the limited and open domain corpora. On the one hand, we automatically selected a set of sentences using a multi-leveled token search method described in [20]. On the other hand, a set of sentences were manually designed by a linguist, in order to cover a set of relevant linguistic units that could not be observed in the automatically selected sentences.

Open Domain. The design of speech corpora for our open domain voices was largely inspired by the language resource specification used in the TC-STAR project [4]. Thus, it started by automatically selecting candidate sentences from a large collection of newspaper articles in order to cover, as much as possible, a set of linguistically-motivated units,[2] so that even in the most unfavorable contexts, the TTS can render a speech signal with good enough quality.

The frequency that each linguistic unit can occur is highly dependent upon the text corpus. Moreover, in [17] the correlation coefficient between the frequency distributions of the triphones shared by two corpora was found not to reach values above 0.3. Therefore, covering only the most frequently observed units is not a solution, unless we are designing a corpus for a limited domain voice. Then our decision was to let the automatic selection algorithm cover all the units. However, since a complete coverage of such units cannot be achieved without a prohibitively large sentence set, that would take too much time to record and annotate, the search for relevant text prompts ends when a predefined coverage threshold is reached.

Another issue that must be addressed is the covering of some lexical items that are only observed in specific domains. Even though, such domains (*e.g.* phone numbers, economy, currencies, computer science terms, frequently used foreign names and expressions, typical dishes, touristic attractions, or even countries and their capitals) are sometimes so relevant for the daily use of the language that, at least, their most frequent lexical items are likely to be typed by the end users. On the other hand, the naturalness of speech is highly dependent on a set of linguistic characteristics that combined with purpose and context may convey distinct effects. Therefore, the manually designed prompts should account for several features, namely, a list of the most frequent verbs in EP in both first and second persons. Hence, we started by computing the occurring frequency of each verb lemma in a corpus of around 1,600,000 newspapers' articles, based on the results of a morpho-syntactic tool described in [10]. The human-computer interactions strongly benefits from the use of spontaneous prompts frequently observed in our daily conversations. Therefore, previously recorded human dialogs in the CORAL corpus [16] were orthographically transcribed and subsequently recorded by the speakers.

Another set of prompts was built with the purpose of providing the speech inventory with additional linguistic events, namely filled pauses and prolongations of segmental material in predictable locations and with different functions (changing a subject, preparing the subsequent units, taking the floor and also as mitigating devices) [8], as well as conversational grants, (*e.g. hum*) with different values. The manually designed prompts also account for all types of interrogative sentences and declarative sentences with the same lexical material as a yes/no question, as it is well known that intonational contours varies distinctly according to the sentence type.

Limited Domain. The design of limited domain speech resources was carried out as follows. Firstly, we gathered a large set of domain-specific sentences.

[2] Diphones, triphones and syllables.

Then, we followed the frequency-dependent[3] approach used in the LC-STAR project [22] while deciding which words should be included in the word lists. After defining the set of words and word sequences to be covered by the automatic procedure, the sentence selection starts. Finally, additional sentences are manually designed in order to provide the corpora with the most relevant words in appropriate contexts, if such words and contexts were not found in the automatically selected prompts.

2.2 Phonetic Segmentation and Multi-level Utterance Descriptions

The phonetic segmentation of the databases is performed in three different stages. Firstly, the speech files are segmented by a hybrid speech recognizer (*Audimus* [9]) working in forced alignment mode. Next, such segmentations are used by the HTK programs [21] for training context-independent speaker-specific HMMs. The speaker-adapted models are subsequently provided to a phonetic segmentation tool based on weighted finite state transducers allowing for many alternative word pronunciations [11].

The spoken utterances were prosodically annotated following ToBI guidelines.[4] The utterance's orthographic transcriptions are then combined with the respective phonetic segmentations, using a procedure described in [12], in order to obtain a realistic and multi-leveled description of the spoken utterances. Moreover, those descriptions are enhanced by additional descriptions, such as F0 values of the speech signal and prosodic annotations. The F0 values are assigned to the respective phonetic segments based on the temporal inclusion criterion.

3 Linguistic Analysis

3.1 Speaker-Adapted Prosodic Models

The general quality and naturalness of synthetic voices crucially depends on the building of large databases annotated at multiple levels for the training and testing of prosodic models able to generate adequate rhythmic and intonational patterns. One of the most striking difficulties in the building of new voices in the present framework is that the type of annotation required is extremely time consuming and the models trained for one voice or speaking style are most often inadequate for another. Models trained on a laboratory corpus of read texts or elicited sentences by non-professional speakers, for instance, may hardly be used to build a voice based on new databases recorded by professional ones, as strong mismatches are found both for the phrasing and tonal assignment strategies used. This clearly affects the selection of the units to concatenate, as often adequate exemplars cannot be found and several discontinuities are introduced.

[3] In limited domain applications, the text prompts do not aim to be an abstract representation of the language use, thus, occurring frequencies carry relevant information for modeling the language in that specific domain.

[4] http://www.ling.ohio-state.edu/~tobi

It is worthwhile to note that most studies based on laboratory corpora recorded by non-professional speakers present EP as a language with sparse accentuation and just one level of phrasing (e.g. [19]) whereas it is clear from available data of spontaneous speech and professional reading that at least two levels of phrasing are needed to improve our results. Although the basic pitch accent inventory is in agreement with laboratory studies, it is also mandatory to be able to account for rather common pitch accents in our data that are absent or not well enough represented in laboratory corpora (*e.g.* L+H*, in nuclear position, the most frequently used in association with new information or old information that need to be reactivated or ^H*, consistently used in repetitions for further specification, or to correct given or inferable information).

Most of our effort in what prosody is concerned, has thus been dedicated to accelerate the annotation process by training statistical models for the automatic feature extraction, in order to reduce as much as possible the need for manual intervention. So far, we have been mainly concerned with the improvement of prosodic phrasing models. Those are essential to achieve better results in what tonal scaling is concerned. On the other hand, as the annotation scheme is closed to the English ToBI one, and the phonetic correlates of each type of tonal event for EP are relatively well known, we expect to be able to reduce the number of errors in the automatic tagging of such events. Drawing on previous work in the line of [6], a new database for a professional speaker was automatically parsed with a CART trained and tested on text based annotations, only [18]. In spite of *break/no-break* decisions produced correct results in only 70% of the cases, the manual correction was considerably facilitated. The use of manually annotated data for training prosodic phrasing models accounting for the speaker-specific reading strategies has proved to be worthwhile, as the adapted *break/no-break* detection models reached substantially higher performances (precision=88.44%; recall=93.90%).

3.2 Grapheme to Phone

The current G2P component follows the same approach of the Festival system comprising a lexicon, an addenda and a set of classification trees, one for each symbol of the alphabet. To train the classification trees a rather large lexicon is required. The size of the lexicon depends on the language and on its regularity. In our case we used an EP lexicon with around 80,000 entries. Each lexicon entry includes the word orthography, a part-of-speech (POS) tag and the corresponding sequence of phones with a lexical stress mark on the central vowel of the stressed syllable. The orthography and phonetic transcription must be aligned so that each letter corresponds to a single phonetic symbol. This symbol can represent a single phone, a sequence of phones or no phone at all. The goal of a classification tree is to predict which is the phonetic symbol associated with a given letter of the alphabet in a specific context and for a word with a given POS tag. The context must have a finite length that needs to be optimized for each language. In the case of EP we achieved better results by using 3 letters to the left and 6 to right of the letter being transcribed. This technique produced

93.28% and 99.12% accurate transcriptions in the test set (10% of the full lexicon) at the word and grapheme levels, respectively. The performance measures for the full 80K lexicon were 3.71% and 0.48% for word and phone error, respectively. These results are slightly worse (around 2%) than the ones that we have achieved using phonetically motivated rewriting rules. This approach, however, has the advantage of being automatically trainable and thus easily extendable to other languages.

The system lexicon must include all the words that are not correctly transcribed by the classification trees. For performance reasons, however, we have decided to include also the most frequently used words in EP. The third element of the G2P component is the addenda. It works as an exception lexicon in which the user can override the way the system reads certain words. The addenda can be particularly useful for non-standard words like company names or even foreign words that are not correctly pronounced by the system.

4 System Architecture

4.1 Text Splitter

The input text of a speech synthesizer can have a wide range of variability. Moreover, one may assist to a dramatic degradation of the system's overall performance when long sentences, paragraphs and text documents are processed as a single unit. These large text chunks require longer processing that delay the generation of the audio output. The input text is split into sentences based on its punctuation in order to minimize that delay. However, punctuation marks can be mistakenly parsed (*e.g.* dots are not used solely for sentence breaks, they can also be used in abbreviations, numbers and even dates). A solution for this is to require that the full stop is followed by a space or a capitalized word. Such restrictions handle the most cases like numbers and dates but not the abbreviations, which are addressed as follows. A large abbreviation inventory is built in advance to enable the system to spot such tokens within the input text. The abbreviation identification can help in distinguishing the dot from a full stop, but the system must also take into account that some abbreviations can occur at the end of sentences (*e.g.* etc.).

4.2 Text Normalizer

The text normalizer is responsible for rearranging text in a normalized form, so that the following components can be more effective. It is a task that requires constant maintenance. Moreover, conventions are useless when it comes to deal with general normalization problems, since there are many writing conventions for similar contexts. For example, numbers in a certain language can have different convention domains, like economical and scientific domains. This is a strong hit in any attempt to design *general* text normalization method. Besides, addressing so many ambiguities can make the system inflexible in the presence of new paradigms.

Considering a general approach, Dixi addresses the normalization problem in two distinct steps. Firstly, text tokens are tagged according to their syntactic form. For example, the token *"1234,34"* is tagged as a number, whereas the token *"76-12-01"* is marked as a date (according to traditional European Portuguese standards). The real text normalization only takes place in the following step, since token to word rules are applied only there. Hence, with all tokens already tagged, specific modules are then applied to carry out the necessary conversions. However, this solution is just a course of action, it does not solve the problem as a whole. In order to increase the system flexibility, both identification and modification levels were categorized according to *Language*, *Region* and *Domain*. The categorization of the normalizer levels minimizes ambiguity, and enables the users to parameterize the normalizer so that it can meet their own demands.

4.3 Part-of-Speech Tagging

The fundamentals of the POS tagger currently used in Dixi was described in [13]. It consists of a lexicon comprising around 22,000 orthographic forms, containing a pair list in the form of *tag/probability* each. The lexicon is used along with a POS tri-gram grammar in order to find the most likely POS tag sequence, whose tags are subsequently assigned to the respective words.

4.4 Prosodic Phrasing

Voice-specific word break models enconded in the form of CARTs were trained as described in 3.1. In run time, the prosodic phrasing is performed making use of the model specifically trained for the currently selected voice.

4.5 Phonological and Phonetic Descriptions

As soon as the word list is grammatically tagged, the pronunciation generation is triggered. This procedure consists of the following steps. Given a particular word, a user-supplied lexicon addenda, if any, is searched for a that written form with a matching grammatical tag. When a search is well-succeeded, pronunciation generation procedure is finished and the respective phonetic sequence is used. If no such entry is found, the procedure resumes by searching for the word pronunciation in the lexicon. Finally if the word is still not present in the lexicon, not even with another grammatical tag, the pronunciation is generated by a set of CARTs, trained as described in 3.2. Up to this stage, pronunciations were generated for isolated words. However, post-lexical phonological processes play an important role in connected speech. Hence, a set of post-lexical rules is then applied to address that problem and produce more realistic utterance descriptions at this level.

4.6 Acoustic Synthesis

Unit Selection Synthesis. The waveform generation is based on a multi-level version of the *cluster unit selection* algorithm [2] and will be further described in

a future paper. Our unit selection algorithm makes use of phone target durations to discard durational outliers in run time, rather than adjusting the durations of the selected units. Therefore, despite the local signal modifications carried out in order to soften the transitions at the concatenation points, the system uses the speaker-specific prosody, available in the recordings. Moreover, even though phonetic segments constitute the basic acoustic units used by *Dixi*, they are searched in a top-down fashion, in order to first search for candidate units coming from the most appropriate prosodic and phonetic contexts (e.g. phone belonging to a word in a specific position, or a syllable in a specific position, or triphone, or a diphone, etc.).

Parametric Synthesis. A parametric approach, based on the HMM synthesis, can also be used within the Dixi system. Such an approach automatically draws a correlation between acoustic features and a set of symbolic features derived from the input text. The training procedure of the HMMs is carried out by the HTS [15] Toolkit. Using tree-based context cluster HMM models, HTS extracts spectral information, average F0 and a voiced/unvoiced decision every 5ms. The features utilized by the HMM synthesizer as well as the context clustering questions, will be described in a future paper. The speech signal is generated using the MLSA[5] filter, proposed in [7].

5 Conclusions

We have described the development of a new text-to-speech system for EP. Besides, a strong emphasis was also put in the description of the language resources needed by the system, as well as the training methods used in the linguistic analysis of the input text. Finally, we described how the system can be parameterized to meet the user-specific requirements.

Acknowledgments

This work was funded by PRIME National Project TECNOVOZ number 03/165.

References

1. Black, A.W., Lenzo, K.A.: Flite: a small fast run-time synthesis engine. In: SSW4 (2001)
2. Black, A.W., Taylor, P.: Automatically clustering similar units for unit selection in speech synthesis. In: Eurospeech 1997 (1997)
3. Black, A.W., Taylor, P., Caley, R.: The Festival Speech Synthesis (2002)
4. Bonafonte, A., Hoge, H., Kiss, I., Moreno, A., Ziegenhain, U., Heuvel, H., Hain, H., Wang, X., Garcia, M.: TC-STAR: Specifications of language resources and evaluation for speech synthesis. In: LREC 2006 (2006)

[5] Mel Log Spectral Approximation.

5. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Chapman and Hall, Boca Raton (1984)
6. Hirschberg, J., Prieto, P.: Training intonational phrasing rules automatically for english and spanish text-to-speech. Speech Communication 18 (1996)
7. Imai, S.: Cepstral analysis synthesis on mel frequency scale. In: ICASSP- 1983 (1983)
8. Moniz, H., Mata, A.I., Viana, C.: On filled and prolongations in european portuguese. In: Interspeech 2007 (2007)
9. Neto, J.P., Meinedo, H.: Combination of acoustic models in continuous speech recognition hybrid systems. In: ICSLP 2000 (2000)
10. Oliveira, B., Pona, C., Matos, D., Ribeiro, R.: Utilização de xml para desenvolvimento rápido de analisadores morfológicos flexíveis. In: XATA 2006 - XML: Aplicações e Tecnologias Associadas (2006)
11. Paulo, S., Oliveira, L.: Generation of word alternative pronunciations using weighted finite state. In: Interspeech 2005 (2005)
12. Paulo, S., Oliveira, L.C.: MuLAS: A framework for automatically building multitier corpora. In: Interspeech 2007 (2007)
13. Ribeiro, R.D., Oliveira, L.C., Trancoso, I.M.: Using morphossyntactic information in tts systems: Comparing strategies for european portuguese. In: Mamede, N.J., Baptista, J., Trancoso, I., Nunes, M.d.G.V. (eds.) PROPOR 2003. LNCS, vol. 2721. Springer, Heidelberg (2003)
14. Taylor, P., Black, A.W., Caley, R.: Heterogeneous relation graphs as a formalism for representing linguistic information. Speech Communication 33 (2001)
15. Tokuda, K., Zen, H., Black, A.W.: An HMM-based speech synthesis system applied to english. In: 2002 IEEE SSW (2002)
16. Trancoso, I., Viana, C., Duarte, I., Matos, G.: Corpus de dialogo CORAL. In: PROPOR 1998 (1998)
17. van Santen, J.P.H., Buchsbaum, A.L.: Methods for optimal text selection. In: Eurospeech 1997 (1997)
18. Viana, C., Oliveira, L.C., Mata, A.I.: Prosodic phrasing: Machine and human evaluation. Speech Technology 6 (2003)
19. Vigário, M., Frota, S.: The intonation of standard and northern european portuguese. Journal of Portuguese Linguistics 2(2) (2003)
20. Weiss, C., Paulo, S., Figueira, L., Oliveira, L.C.: Blizzard entry: Integrated voice building and synthesis for unit-selection tts. In: Blizzard 2007 (2007)
21. Young, S., Evermann, G., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: The HTK Book (for HTK Version 3.2.1) (2002)
22. Ziegenhain, U., Hoge, H., Arranz, V., Bisani, M., Bonafonte, A., Castell, N., Conejero, D., Hartikainen, E., Maltese, G., Oflazer, K., Rabie, A., Razumikin, D., Shammass, S., and Zong, C.: Specification of corpora and word lists in 12 languages. Report 1.3, Siemens AG (April 2003)