SIMPLIFICATION OF TTS ARCHITECTURE VS. OPERATIONAL QUALITY

Eric Keller

Laboratoire d'analyse informatique de la parole (LAIP)
Faculté des Lettres, Université de Lausanne, Switzerland
eric.keller@imm.unil.ch

ABSTRACT

Many applications in mobile telephony and portable computing require high-quality speech synthesis systems with a very modest computational footprint. Our text-to-speech system for French gives satisfactory performance in phonetisation and prosody with considerably reduced computational resources. Using the Mons (Belgium) diphone data base, the program's current version runs in real time on Pentium-type PCs or Mac PPCs. The code requires 442 k, minimum RAM requirement is 4700 k, the minimum disk requirement is 5560 k. The phonetisation and prosody processing has been brought to a first level of optimal compromise between quality and computational footprint. Major further reductions in space requirements would probably necessitate a re-evaluation of sound generation procedures.

INTRODUCTION

Despite its considerable promise, speech synthesis is still not being used on a wide-scale basis in public service contexts. Wider acceptance of speech synthesis devices will probably depend on three factors: (a) better quality, (b) smaller computational footprints, and (c) more attractive pricing.

With respect to the first point, the linguistic quality of the text-to-speech translation process must be largely error-free, and the prosody and acoustic quality of the final output must be pleasant to listen to, if public TTS acceptance is to increase, and commercial use of TTS systems is to rise significantly over current levels.

A second, economically relevant objective is the simplification of TTS design. Smaller and simpler systems can run faster and can be ported to a larger class of machines, which in turn assists in widening TTS use. Of increasing importance are mobile systems (mobile telephony, TTS-equipped car radios, personal digital assistants [PDAs] capable of speech, etc.). The question thus arises whether significant simplifications in TTS design can be achieved without quality compromise, or even with attendant increases in operational quality.

Work performed in our laboratory over the last five years has addressed this issue. All parts of a complete French-language TTS system have been examined in terms of their likely contribution to the final output quality, and a TTS prototype has been assembled that respects both quality and size criteria.

PHONETISATION

With respect to the "phonetisation" process (grapheme-to-phoneme translation), a maximum of 5 wide-transcription errors per 100 sentences was set as an acceptable criterion. A lesser error frequency was considered desirable.

A first major difficulty in French-language phonetisation concerns the trade-off between grapheme-to-phoneme rules and the use of pronunciation dictionaries. The use of rules is preferred, since they can be used to generate a first attempt at pronunciations of names and neologisms. We found that a core set of 460 grapheme-to-phoneme rules predicted about 86% of the pronunciations in a French-language pronunciation dictionary. The ill-predicted pronunciations are often part of rarely-used words. We therefore identified about 6000 words with exceptional pronunciations that are commonly used. In our

current prototype, these exceptional pronunciations consume just 126 k of RAM, while the rules and the rule application mechanisms consume 22 k of C code.

A second major difficulty concerns such phonological rules as liaison ("deux pas" /dø/, but "les deux autres" /døz/) and chaining ("il a" /i la/). As many others, we certainly found liaison rules to be complex, since they relate not only to grammatical proximity, but also to frequency of usage. However, we found that a relatively simple rule took care of the most senior liaison rule. We permit liaison (if at all) within a prosodic group, and do not permit liaison across prosodic group boundary (prosodic groups based on Keller et al., 1993).

A third major difficulty in this area involves homograph-heterophones (HHs). French turned out to be a relatively benign language in this respect, since we identified less than 100 HHs in reasonably frequent use. Some fall into classes ("président" - "détergent"; "acceptions" - "détections"; "reporter" - "supporter"). For each class as well as for each exceptional case, we specified local grammatical contexts to attempt to disambiguate the homography.

A fourth difficulty has to do with number reading. Not only are there different types of reading traditions for the numbers 70-99 (French, Belgian, Swiss-Geneva, and Swiss Romand), but there are also different traditions of reading off the decimals (singles and doublets). Furthermore, special care must be taken with zero (0 = "zéro", but 101 = "cent un" [no zero pronounced]), and exceptional pronunciations (9 éléphants /nøf/, but 9 heures /nøv/). Our solution to this problem was to simply hard-code the entire number system, as well as its various exceptions.

The fifth and final difficulty concerns the pronunciation of names. Since this problem is literally without limit, our solution has been to provide an external dictionary that can easily be extended by the user. The potential size of the dictionary has no intrinsic limit in either speed of access or size. As it grows, it will simply enlarge the program's RAM requirements.

The result of the various design decisions presented here is somewhat akin too "lossy compression" in image processing (e.g. JPEG). By far most precision is maintained, but certain cases cannot be disambiguated or interpreted correctly. For example the final consonant of "tous" is not identified correctly in the sentence "Nous regarderons tous les jeunes journalistes" (the "s" should be pronounced), since our local grammar is to simplistic. Similarly, there should be an impediment to the liaison between "vous" and "a" in the sentence "L'homme qui était avec vous a vu l'accident". Our "lossy" synthesis system does not identify this grammatical structure correctly, and thus allows the liaison.

The crucial question therefore is how well the system performs on general, non-selected text. In the context of the current AUPELF evaluation of French-language speech synthesis devices, our system was recently tested for wide-transcription errors. As test material, we took the last 500 sentences of a 3934-sentence, 32'887-word text called "Le mot et l'idée", produced by the LIMSI, Orsay, France. On these 500 last sentences, we identified 1.34 errors per 100 sentences. We thus consider that we have attained and surpassed our initial quality criteria, despite the simplifying assumptions we have incorporated into the system.

PROSODY

In reviewing the system's processing requirements for prosody, it was again found that significant simplifications could be made. In a series of studies on the prediction of timing structures related to prosodic phrases, it was found that a relatively simple, modified Grosjean-type algorithm made systematically better predictions for timing than several syntactically-derived algorithms (Keller et al., 1993; Keller & Zellner, 1995, 1996; Zellner, 1994, 1996a, 1996b). On the basis of this algorithm we identify so-called "performance structures" in speech sequences. These are the building blocks for our prosodic processing.

We then examined in a long series of experiments which segmental, syllabic and

phrase-level factors contributed the greatest explanation of variance to a general linear regression solution for segmental duration. The following factors were found to explain a total of around 48% of the variance: (a) the durational class of the segment preceding the current segment, (b) the durational class of the current segment, (c) the durational class of the subsequent segment, (d) the durational class of the ulterior segment, (e) the position in the prosodic group of the syllable containing the current segment, (f) the grammatical status of the word containing the current segment, and (g) the number of segments in the syllable containing the current segment. "Durational class" refers to one of nine clusters of typical durations for segmental duration. The coefficients from this statistical solution can be used as a highly efficient and reasonably detailed predictor of timing information.

For the calculation of F0, we similarly chose a relatively simple, but powerful approach, Fujisaki modelling (for details on this design choice see Keller *et al.*, in press). As modified for use with French in our laboratory (e.g., we model every syllable, not just stressed syllables), Fujisaki modelling gives quite satisfactory intonation patterns for declarative sentences. Here again, the quality - simplicity trade-off implemented in our system provides a fairly satisfactory solution for the large majority of declarative sentences. At the same time, certain interrogative sentences as well as certain types of interactive speech may be less well modelled by Fujisaki methods. However, they typically represent a numerically less important subset of typical TTS applications.

Since it is difficult to design convincing tests of prosody, the best judgement of the prosodic performance of our system is probably obtained by listening to the sound examples furnished with this article.

PERFORMANCE

The performance of LAIPTTS was examined with respect to three criteria, speed, size, and intrinsic limits.

The system's *speed* is considered satisfactory, since it is real-time on contemporary desktop computers (Pentium-PCs and PPC-Macintoshes) (Figure 1). Real-time performance is impeded only when a exceedingly short sentence precedes a very long one. In this situation, the reproduction time is slightly inferior to the calculation time.

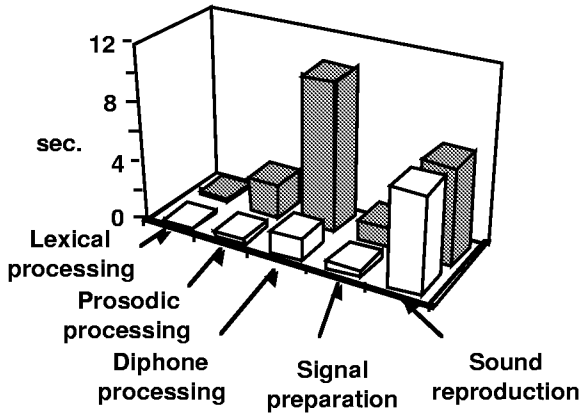

Figure 1: Calculation speed for a typical 7-second sentence. white: PPC-80 MHz, grey: 68k-fpu-33 MHz. Diphone processing could be accelerated if the diphone base were taken into RAM.

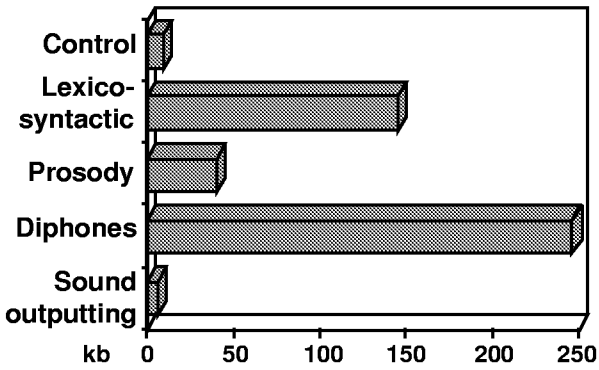

Figure 2: Distribution of code segments in LAIPTTS (out of a total of 442k). Diphone processing is quite burdensome, as is lexico-syntactic processing.

In terms of *space*, the system was evaluated with respect to RAM and disk space. Although these are obviously overlapping and mutually exchangeable types of space, a listing of current requirements gives some idea of obligatory space occupation (Figure 2, Tables I and II). It can be seen that the space requirements are quite modest when compared to the typical 15-30 Mb-sized high-quality TTS

systems developed for various telecommunications firms.

The *intrinstic limitations* of the system are primarily found in the areas of the "finer processing" for speech. For example, the system makes no semantic analysis, and thus cannot emphasise words or passages that need particular prosodic emphasis. Similarly, there is obviously no pragmatic analysis, and prosodic markings that seem obvious to humans on the basis of context are not at all obvious to the TTS system. To handle these requirements in the future, we are intending to develop an external marking system that permits users with sophisticated analysis systems to use LAIPTTS in a contextually appropriate fashion.

Table I: RAM Space

Code	442k
Structures and working space (min.)	4255k
Total	**4697k**

Table II: Disk Space

Application	481
Exceptional pronunciations	130
Abbreviations and fixed expressions (min.)	37
Proper names (min.)	37
Statistical data base	37
Diphone data base	4700
Preferences	37
Sound signal (typical)	100
Total	**5559 k**

CONCLUSION

Our current prototype (using the Mons, Belgium, "Mbrola" diphone system) is characterised by high computational efficiency, small size, and good acoustic quality. The core application is some 450 kb in size, and operation is real-time on mid-sized Pentium- or PPC-RISC type of computers. At the same time, phonetisation errors remain well below the initial acceptability threshold. Timing predictions at the segmental levels reach the .7 correlation level. As yet non-normed listening tests document good perceptual and acceptability characteristics.

Current work focuses on the completion of the system by the creation of a laboratory-internal diphone base and diphone motor, and on developing a number of perceptual quality tests in collaboration with AUPELF and our industrial partners. Further planned work is set to examine extensions required to handle wider linguistic materials, such as dialogues and interrogative sentences.

FILES FURNISHED ON CD ROM

We provide two files with this article, **met_nat.wav**, and **met_syn.wav**. The first is a recording of a natural speaker giving a standard Swiss weather report. The second is a reproduction of the same text with LAIPTTS. It can be seen that LAIPTTS preserves a speech rhythm typical of French declarative sentences.

REFERENCES

Keller, E., & Zellner, B. (1995). A statistical timing model for French. *XIIIth International Congress of the Phonetic Sciences, 3* (pp. 302-305). Stockholm.

Keller, E., Zellner, B., & Werner, S. (in press). Improvements in Prosodic Processing for Speech Synthesis. Proceedings of COST Workshop, Rhodos, Greece (1997).

Keller, E., & Zellner, B. (1996). A timing model for fast French. *York Papers in Linguistics*, 17, University of York. 53-75.

Keller, E., Zellner, B., Werner, S., and Blanchoud, N. (1993). The prediction of prosodic timing: Rules for final syllable lenthening in French. *Proceedings ESCA Workshop on Prosody, September 27-29. Lund, Sweden.* 212-215.

Zellner, B. (1994). Pauses and the temporal structure of speech, in E. Keller (Ed.) *Fundamentals of speech synthesis and speech recognition.* (pp. 41-62). Chichester: Zellner, B. (1996a). Structures temporelles et structures prosodiques en français lu. *Revue Française de Linguistique Appliquée: La communication parlée. 1.* (pp. 7-23). Paris.

Zellner, B. (1996b). Relations between the Temporal and the Prosodic Structures of French, a Pilot Study. 3thd Joint Meeting ASA & ASJ, Honolulu.