International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:11, 2009

Unit Selection Algorithm Using Bi-grams Model For Corpus-Based Speech Synthesis

Mohamed Ali KAMMOUN and Ahmed Ben HAMIDA

Abstract—In this paper, we present a novel statistical approach to corpus-based speech synthesis. Classically, phonetic information is defined and considered as acoustic reference to be respected. In this way, many studies were elaborated for acoustical unit classification. This type of classification allows separating units according to their symbolic characteristics. Indeed, target cost and concatenation cost were classically defined for unit selection.
In Corpus-Based Speech Synthesis System, when using large text corpora, cost functions were limited to a juxtaposition of symbolic criteria and the acoustic information of units is not exploited in the definition of the target cost.
In this manuscript, we token in our consideration the unit phonetic information corresponding to acoustic information. This would be realized by defining a probabilistic linguistic Bi-grams model basically used for unit selection. The selected units would be extracted from the English TIMIT corpora.

Keywords—Unit selection, Corpus-based Speech Synthesis, Bi-gram model

I. Introduction

CONCATENATIVE Text-To-Speech synthesizers join pre-recorded segments of speech data in order to produce high quality output speech [1,2]. The synthesizer has to find the best unit to concatenate from an inventory of speech material.
Before unit selection, concatenative synthesis involved concatenation of units (usually diphones) from fixed databases, i.e. databases which contained only one example of each unit. However, having only one example of each unit in the database can not account for variation in pronunciation generally found in natural speech. Segmental co-articulation effects spread, as it is generally known, also across more than one phone or diphone. Additionally, prosodic factors like stress, position within the syllable or intonational phrase affect the pronunciation of a unit. Correct prosody is achieved here by signal processing techniques which distort the waveform and impair the quality of the output. Also high frequency of unit concatenation points proved to affect the quality of speech, since it resulted in more audible joins between the units. The primary motivation for unit selection synthesis was to improve synthesis quality by reducing spectral mismatches at the points where units are concatenated. This is achieved by storing multiple examples of a unit recorded in different phonetic and prosodic contexts in the database, and choosing the proper unit for the given context, automatically, at synthesis time.
In this way, Corpus-Based Speech Synthesis (CBSS) was introduced. In fact, Corpus-based concatenative approach to speech synthesis has been widely explored in the research community in recent years. In this approach, best sequences of phone or subphone-sized units are chosen from a large inventory of possible units to synthesize input text, by minimizing the overall cost function. The overall cost is often modelled as the weighted sum of target costs and concatenation costs on the various features such as spectral, intonational and duration features.
In the new corpus-based speech synthesis framework that we present in this paper, we go further and propose a probabilistic approach to unit selection in concatenative speech synthesis.
We are pursuing this approach in the hope that a probabilistic approach will make it easy to establish a method that is mathematically manageable, needs fewer tuning parameters, and is easy to train, by taking advantage of statistical properties emerging from the data. It can be regarded as a more constrained subclass within the larger class of general cost-based approach.
The paper outline is as follows. First, we considered relevant background on topics in unit selection algorithm. Next, we introduce our probabilistic framework for unit selection. It is followed by the descriptions of the target and concatenation models in our probabilistic approach. We then briefly describe the unit search mechanism after that. We then describe the way we generate the target word sequence from input. We finally describe the implementation with TIMIT English corpora followed by conclusion.

II. Unit Selection Algorithm: State of the Art

In the context of concatenative speech synthesis, unit selection algorithm are first proposed in [3] and used in the speech synthesis system CHATR and the systems influenced by it. The starting point of the unit selection algorithm is a database of N units u_i and a sequence of T target units t_T. The unit selection algorithm finds the units from the database that best match the given synthesis target units. The quality of the match is determined by two distance functions, expressed as costs[4, 5, 6, 7, 8, 9, 10, 11, 12]: The target cost C^t corresponds to the perceptual similarity of the database unit u_i to the target unit t_T. It is given as a sum of p weighted individual feature distance functions C_k^t as:

$$C^t(u_i, t_T) = \sum_{k=1}^{p} w_k^t C_k^t(u_i, t_T) \quad (1)$$

The concatenation cost C^c predicts the discontinuity introduced by concatenation of the unit u_i from the database with

Manuscript received November 2008.
M.A. KAMMOUN is with the Research Unit in Information technology and Medical Electronics,(TIEM),(ENIS), BP W, 3038 SFAX, TUNISIA, email:MohamedAli.Kammoun@isetma.rnu.tn.
A.B. HAMIDA is the Director of the Research Unit in Information technology and Medical Electronics,(TIEM), National School of Engineers,BP W, 3038 SFAX, TUNISIA email:Ahmed.Benhamida@enis.rnu.tn.

a preceding candidate unit u_{i-1}. It is given by a weighted sum of q feature concatenation cost functions C_k^c:

$$C^c(u_{i-1}, u_i) = \sum_{k=1}^{q} w_k^c C_k^c(u_{i-1}, u_i) \quad (2)$$

Consecutive units in the database have a concatenation cost of zero. Thus, if a whole phrase matching the target is present in the database, it will be selected in its entirety. The unit selection algorithm has to find the least costly path that constitutes the target. Using the weighted target cost $w^t C^t$ as the state occupancy cost b_i, and the weighted concatenation cost $w^c C^c$ as the transition cost a_{ij}, the optimal path can be efficiently found by a Viterbi algorithm [13, 14].

III. Probabilistic approach to unit selection

In a speech synthesis framework where units are selected from the corpus, we are given some input specification such as specifications for phone-sized or even finer subphone units, $s = s_1, ..., s_N$. The major work of the synthesizer is to find a best sequence of units $u = u_1, ..., u_N$ for this input specification. A specification for a unit s_i can be a collection of target features, $s_i = (f_i(1), ...f_i(p))$. These features may include such things as a phone label, a duration target, and an F_0 target for the $i-$th unit.

In our proposed method, words corpus can be seen as a fully connected state transition network through which the unit selection algorithm has to find the most probably path that constitutes the target.

In fact, we observed that the database can offer phonemic information for every word in the corpus. The first step can be presented as follow: we browsed all directories to create a list presenting basic words and corresponding phonetic transcriptions.

As second step, we associated for every word a phonetic transcription. Given sentence to synthesis, this implies that we have to make a selection on all the phonetic transcriptions proposed by the previous step. On standard way of doing this is by the probabilistic linguistic model "$N - grams$" [15, 16, 17].

The $N - grams$ model appeared in the context of speech recognition to estimate the sequence of the words $w_1, , w_N$ in a given language.

In the context of pre-selection for a given sentence $W = (w_1, w_2, ..., w_N)$, we look for the best sequence of transcription $\hat{T}$ on all possible sequences $T = (t_1, t_2, ..., t_N)$ made on all the phonetic transcriptions $\{ph_1, ph_2, ..., ph_M\}$:

$$\hat{T} = \arg\max_T p(T|W) \quad (3)$$

By Bayes's rule, this is equivalent to find:

$$\hat{T} = \arg\max_T \frac{p(T, W)}{p(W)} = \arg\max_T \frac{p(W|T)p(T)}{p(W)} \quad (4)$$

The denominator of the second equation is independent of T, we can ignored in the search of $\hat{T}$.

The $N - grams$ for pre-selection assumes the following estimation:

- The probability of a word, knowing the one who precedes it, depends on the transcription.
- The probability of a transcription, knowing the one who precedes it, depends on $n-1$ previous transcriptions.

As result:

$$\begin{aligned} p(W|T) &= p(w_1, w_2, ..., w_N|t_1, t_2, ..., t_N) && (5)\\ &= p(w_1|t_1, t_2, ..., t_N)... \\ &.. \quad p(w_N|w_1, ..., w_{N-1}, t_1, t_2, ..., t_N) && (6)\\ &\approx \prod_{i=1}^{N} p(w_i|t_i) && (7)\\ p(T) &= p(t_1, t_2, ..., t_N) && (8)\\ &= p(t_1)p(t_2|t_1)...p(t_N|t_1, t_2, ..., t_N) && (9)\\ &\approx \prod_{i=1}^{N} p(t_i|t_{i-1}, t_{i-2}, ..., t_{i-N+1}) && (10) \end{aligned}$$

It is then straightforward to see the problem in terms of a finite state automaton. This machine represents a model Bi-grams where $n = 1$ [15 , 16, 17]. The model Bi-grams considered is represented by states associated to possible phonetic transcriptions (a state by transcription). For every transition, we associate a probability $p(ph_i|ph_j)$ which represents the probability that a word with the transcription "ph_j" will be followed by a word with the transcription "ph_i". The emission probability $p(w_i|ph_j)$ represents the probability that the phonetic transcription "ph_j" corresponds to the word "w_i".

Once emission and transition probabilities are estimated, final step consists in obtaining the best sequence of words for a given sentence.

In fact, by analogy with cost function, target cost C^t can be replaced by the inverse of the emission probability:

$$C^t = \frac{1}{p(w_i|ph_j)} \quad (11)$$

And the concatenation cost C^c by the inverse of the transition probability:

$$C^c = \frac{1}{p(ph_i|ph_j)} \quad (12)$$

This corresponds to finding the best path in a lattice. One could obviously obtain the best path by first computing the probability of all possible sequences and then retaining the one with highest probability. This is sally a time-consuming task, since the number of possible sequences of tags for a sentence is the product of the numbers of possible tags for all its words. We used a brute-force method for finding all possible paths in a lattice and another algorithm for using it in the context of our bigram model and obtaining the best tag sequence.

IV. Experiments and results

A. Experiment corpora: TIMIT Corpus

The TIMIT Corpus is an acoustic and phonetic database dedicated mainly to speech recognition. It contains the recordings of 630 American speakers, distributed on ”8” regional dialects (”dr1” to ”dr8”). Each one pronounces 10 sentences. These sentences are distributed on three groups:
- sentences of calibration, pronounced by all speakers, serving for illustrating the regional variations (identified as ”sa1” and ”sa2”);
- sentences are drawn lots among 450 well-calibrated phonetic sentences (identified from ”sx3” to ”sx452”);
- sentences are chosen to maximize the acoustic contexts; every sentence is pronounced only once, with a total of 1890 sentences for 630 speakers (identified from ”si453” to ”si2345”).

The total vocabulary of the database is 6100 words. The text is read under good recording conditions. 630 speakers of the base (438 men and 192 women) are distributed between training's folder (462 speakers: 326 women and 136 men) and test's folder (168 speakers: 56 women and 112 men). Every speaker is identified by one letter indicating his genre (”m” for men and ”f” for women). The hierarchical organization of the constituent files' corpus represents its key-point. In fact, data are organized according to the following path : The

```
/<CORPUS>/<USAGE>/<DIALECT>/<SEX>..
..<SPEAKER_ID>/<SENTENCE_ID>.<FILE_TYPE>
With:
    CORPUS :=timit
    USAGE :=train | test
    DIALECT :=dr1 | dr2 | dr3 | dr4 | dr5 | dr6 | dr7 | dr8
    SEX:=m | f
    SPEAKER_ID :== <INITIALS> <DIGIT>
With,
    INITIALS :=3
    DIGIT := 0-9
    SENTENCE_ID :==<TEXT_TYPE><SENTENCE_NUMBER>
With,
    TEXT_TYPE:==sa | si | sx
    SENTENCE_NUMBER :== 1 ...2342
    FILE_TYPE:==wav | txt | wrd | phn
```

TIMIT corpus includes several files corresponding to each sentence. In addition to a speech waveform file (.wav),three associated transcription files (.txt, .wrd, .phn) exist. These associated files have the following form:

```
<BEGIN_SAMPLE> <END_SAMPLE> <TEXT><new-line>
    .
    .
    .
<BEGIN_SAMPLE> <END_SAMPLE> <TEXT><new-line>
where,
  BEGIN_SAMPLE :== The beginning integer sample number for the segment
  END_SAMPLE :== The ending integer sample number for the segment
  TEXT :== <ORTHOGRAPHY> | <WORD_LABEL> | <PHONETIC_LABEL>
where,
  ORTHOGRAPHY :== Complete orthographic text transcription
  WORD_LABEL :== Single word from the orthography
  PHONETIC_LABEL :== Single phonetic transcription code
```

Each extension can be presented as below:
- .wav: SPHERE-headered speech waveform file.
- .txt: Associated orthographic transcription of the words the person said.
- .wrd: Time-aligned word transcription. The words' boundaries were aligned with the phonetic segments using a dynamic string alignment program.
- .phn: Time-aligned phonetic transcription.

B. Lexicon Creation

Lexicon creation from the TIMIT corpus is made by browsing the database by accessing files ”.txt” and ”.wrd” and creates a list containing words from database followed by their phonetic transcriptions.
The first three sentences will be shown as follows:

```
>> corpus_init=corpus

corpus_init =
    'the'          'dhiy'
    'emperor'      'ehmpclprix'
    'had'          'hvaedx'
    'a'            'ix'
    'mean'         'miyn'
    'temper'       'tcltehmpclpaxr'
    ','            ''
    'how'          'hhaw'
    'permanent'    'pclpermixnxehn'
    'are'          'aa'
    'their'        'dhehr'
    'records'      'rehkclkixdcldz'
    '.'            ''
    'the'          'dhix'
    'meeting'      'miydxiyng'
    'is'           'ihz'
    'now'          'naw'
    'adjourned'    'ixdcljherndcld'
    '.'            ''
    .
    .
    .
```

C. Corpus Preprocessing

The TIMIT corpus although it is wide, it is distinguished by simplicity: it does not contain numbers, neither acronyms nor complex proper nouns. Furthermore, the sentences to be synthesized have to contain no spelling mistake. So, the only task which remains to make is to decompose the text into set of states (words and punctuation). The result of this decomposition is shown as follows :

```
>> phrase='she ask if some one needs money'

phrase = she ask if some one needs some money

>> sentence=tts_preprocess(phrase)

sentence =
    'she'
    'ask'
    'if'
    'some'
    'one'
    'needs'
    'money'
```

D. Phonological Analysis

Previously, we presented the advantages of use of the corpus TIMIT as the acoustic database, and we showed that its key-point resides into the hierarchical composition of its files. Let us retain the possibilities offered by the phonetic transcriptions files. Therefore, the phonological analysis of the TIMIT corpus consists in presenting a list of the words corpus associated to their phonetic transcriptions. So, we browse all the directories and all ”.txt” and ”.wrd” files of our corpus to create this list.
This step will create a matrix that presents basic words and

all corresponding phonetic transcriptions.

```
>> corpus_init=corpus;
>> wrd_list=corpus_to_list(corpus_init)
wrd_list =
    ''              {1x1 cell}
    'a'             {1x5 cell}
    'academic'      {1x1 cell}
    'adjourned'     {1x1 cell}
    'all'           {1x2 cell}
    'an'            {1x1 cell}
    'and'           {1x3 cell}
    'answer'        {1x1 cell}
    'appreciated'   {1x1 cell}
    'aptitude'      {1x1 cell}
    'are'           {1x1 cell}
    'as'            {1x1 cell}
    'ashamed'       {1x1 cell}
    'ask'           {1x3 cell}
    .
    .
    .
```

For example, on the previous list, we obtained for the word "ask" three phonetic transcriptions:

```
'aes' 'aeskcl' 'aeskclk'
```

With the same function, we can also get all possible phonetic transcriptions:

```
>> [ph_list,phn_list]=corpus_to_list(corpus_init);
>> phn_list
Phn_list =
    'aa'
    'aal'
    'aan'
    'aenser'
    'aes'
    'aeskcl'
    'aeskclk'
    'aez'
    'ahn'
    'ahnih'
    'aol'
    'aothrihzeyshixn'
    'awdxixdcljh'
    'ax'
    'ax-h'
    'axkclkers'
    'axkclkwihqmixn'
    'axn'
```

The first transcription corresponds to punctuation. The transcriptions 6, 7 and 8 correspond to the word "ask".
Afterward, we created a function which allows us to find phonetic transcriptions corresponding to a given word:

```
>> possible_phn=list_search('ask',wrd_list)
possible_phn =
'aes' 'aeskcl' 'aeskclk'
```

This function will be usefully used to find all possible phonetic transcriptions corresponding to words constituting the sentence to synthesize:

```
>> possible_tags=tts_ph_using_list(sentence,wrd_list)
possible_tags =
    {1x2 cell}
    {1x3 cell}
    {1x1 cell}
    {1x1 cell}
    {1x1 cell}
    {1x1 cell}
    {1x1 cell}
>>possible_tags {:, :}
ans =
'shix' 'shiy'
ans =
'aes' 'aeskcl' 'aeskclk'
ans =
'qixf'
ans =
'sem'
ans =
'wahn'
ans =
'niydcldz'
ans =
'ahnih'
```

E. Unit Pre-selection

At this stage, Bi-grams model was constructed. Before one can use such a model however, one still needs to estimate the relevant parameters.
Computing emission probability is simple. Indeed, this probability represents the number of times when the word "w_i" seems with the phonetic transcription "ph_j" divided by the total number of words with the transcription "ph_j":

$$p(w_i|ph_j) = \frac{\#(w_i, ph_j)}{\#(ph_j)} \tag{13}$$

Similarly, the transitional probability between two transcriptions "ph_j" and "ph_i" represents the number of times when the transcription "ph_i" is preceded by "ph_j" divided by the total number of words with the transcription "ph_j":

$$p(ph_i|ph_j) = \frac{\#(ph_i, ph_j)}{\#(ph_j)} \tag{14}$$

An example of the model Bi-grams is given in figure1.
In order to compute these probabilities, we implement the last two equations (equations 13 and 14) in a function which returns the values of emission and transition probabilities sketched in figure1:

```
>>[emission_probs,transition_probs]=corpus_to_bigrams(corpus_init);
```

"$emission\_probs$" is a $(w \times p)$ matrix where "w" is the total number of words and "p" is the total number of phonetic transcriptions.
Let us note as example columns 6, 7 and 8. These columns correspond to the transcriptions relative to the word "ask":

```
>> emission_probs(:,6:8)
ans =
    .
    .
    .
  0   0   0
  0   0   0
  0   0   0
  1   1   1
  0   0   0
  0   0   0
  0   0   0
    .
    .
    .
```

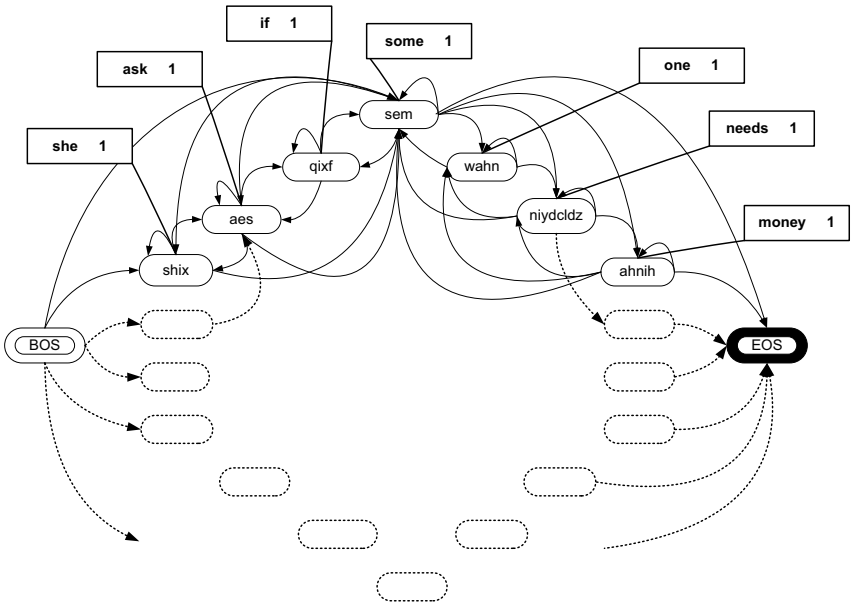

Fig. 1: A possible Bi-grams automaton for TIMIT Corpus (all states are supposed to be fully connected: only a few connections are shown).

The zero probabilities are in the 14^{th} line which correspond to the word "ask" in the matrix "wrd_list". "$transition\_probs$" is a $(p \times p)$ dimension matrix where "p" is the total number of the transcriptions. For this probability, we take the example of the 40^{th} column which corresponds to the phonetic transcription "$dhax$":

```
>> transition_probs(:,40)
ans =
        .
        .
        .
        0
        0
        0
        0.2500
        0
        0
        0
        0
        .
        .
        .
```

A non-zero probability appears in 22^{nd} line. This means that the phonetic transcription "$dhax$" can be followed by the transcription "$bclbaadxaxm$" (22^{nd} line of the matrix "phn_list") with a probability of 0.25. In practice, though, one can never be sure to cover all possible cases in a corpus, however, large it is. People typically address this problem by changing zeros into small non-zero values, which will tend to restrain the algorithm from choosing very unlucky paths, while avoiding the assumption of strict null probabilities. In our script we simply add 10^{-8} to all probabilities [2, 15, 17].

F. Unit Selection

Once emission and transition probabilities are estimated, obtaining the best sequence of words for a given sentence reduces to selecting the best sequence of words transcriptions for the sentence, i.e., the one with highest probability (given the sequence of words and the Bi-gram model). In section 4, we showed that we can replace the target cost by the inverse of the emission probability and the concatenation cost by the inverse of the transition probability. This corresponds to finding the best path in a lattice. As a matter of fact, while figure1 shows a Bi-grams automaton for all possible sentences of TIMIT Corpus, the automaton reduces to a lattice for a given sentence (see figure2).

An example of use, to obtain the best sequence, is presented as follows:

```
>> possible_tags=tts_ph_using_list(sentence,ph_list);
>>tags=tts_tag_using_bigrams(emission_probs,..
..transition_probs,ph_list,tr_phn,sentence,possible_tags)
tags =
        'shix'
        'aes'
        'qixf'
        'sem'
        'wahn'
        'niydcldz'
        'ahnih'
```

In our case, we have 6 possible paths. The adequate path is represented by the matrix "$tags$". Once we found phonetic transcriptions, we browse again the TIMIT corpus and we extract the words samples corresponding to the target transcriptions previously picked.

V. Conclusion and Futures Works

In this paper, we proposed a probabilistic approach to unit selection in concatenative speech synthesis, where all the "costs" are formulated in a probabilistic framework. We have described a new Bigram based approach to Corpus-Based Speech Synthesis. We have given details of how this approach

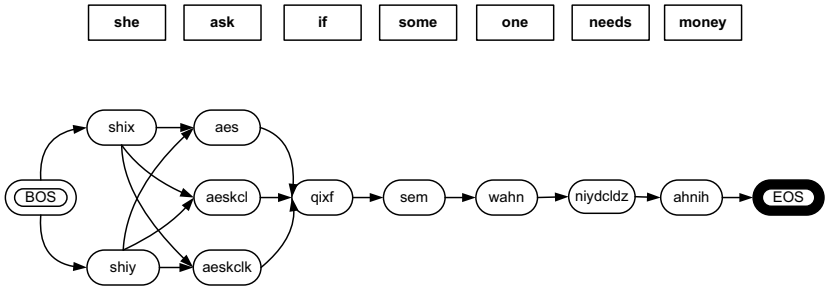

Fig. 2: An example of a lattice Bi-grams for a simple TIMIT sentence. Transition probabilities are associated to arcs.

has been applied to unit selection from large corpora. In fact, the Bigram-based selector algorithm captures phonetic information from large text corpus. The selection is made on computing the best path giving the maximal probability, as opposed to cost functions computation basically used in context of concatenative synthesis. The system is still in its infancy and we plan to improve on various aspects of the system.

APPENDIX A

Bayes' theorem relates the conditional and marginal probabilities of stochastic events A and B:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{15}$$

Each term in Bayes' theorem has a conventional name:
- $P(A)$ is the prior probability or marginal probability of A.
- $P(A|B)$ is the conditional probability of A, given B. It is also called the posterior probability.
- $P(B|A)$ is the conditional probability of B given A.
- $P(B)$ is the prior or marginal probability of B, and acts as a normalizing constant.

There is also a version of Bayes' theorem for continuous distributions. It is somewhat harder to derive, since probability densities, strictly speaking, are not probabilities, so Bayes' theorem has to be established by a limit process. Bayes's theorem for probability densities is formally similar to the theorem for probabilities:

$$f(x|y) = \frac{f(x,y)}{f(y)} = \frac{f(y|x)f(x)}{f(y)} \tag{16}$$

REFERENCES

[1] T. Dutoit (1999). *A Short Introduction to Text-To-Speech Synthesis*. TTS research Team, TCTS Lab.,Faculté polytechnique de Mons, 2004.
[2] J. Schroeter. *Text-To-Speech (TTS) Synthesis*. Circuits, Signals, Speech and Image Processing.
[3] A.J. Hunt and A.W. Black (1996). *Unit selection in a concatenative speech synthesis system using a large speech database*. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Atlanta, GA, pp. 373-376.
[4] T. Toda, H. Kawai, M. Tsuzaki, and K. Shikano (2002). *Unit Selection for Japanese Speech Synthesis Based on Both Phoneme Unit and Diphone Unit*. In Proc. of ICASSP, vol. 1, pp. 465-468, May 2002.
[5] A. Breen and P. Jackson, P. (1988). *Non-Uniform Unit Selection and the Similarity Metric Within BT's LAUREATE TTS System*. 3rd ESCA Int. Workshop, November 1998.
[6] R. Prudon, and C. Alessandro (2001). *A Selection/Concatenation Text-to-Speech System: Databases Development, System Design, Comparative Evaluation*. 4th ISCA Tutorial and Research Workshop on Speech Synthesis, September 2001.
[7] G.R.W. Yi and J. Glass (2002). *Information-Theoretic Criteria for Unit Selection Synthesis*. In Proc. of ICSLP, pp. 2617-2620, September 2002.
[8] M. Lee, D.P. Lopresti and J.P. Olive (2001). *A Text-to-Speech Platform for Variable Length Optimal Unit Searching Using Perceptual Cost Functions*. 4th ISCA Tutorial and Research Workshop on Speech Synthesis, September 2001.
[9] H. Peng, Y. Zhao, and M. Chu (2002). *Perceptually Optimizing the Cost Function for Unit Selection in TTS System With one Single Run of MOS Evaluation*. In Proc. of ICSLP, pp. 2613-2616, September 2002.
[10] R.E. Donovan and E.M. Eide(1998). *The IBM Trainable Speech Synthesis System*. In Proc. of ICSLP, 1998.
[11] T. Nomura, H. Mizuno and H. Sato, H. (1990). *Speech Synthesis by Optimum Concatenation of Phoneme Segments*. 1st ESCA-IEEE Tutorial and Research Workshop on Speech Synthesis, pp. 39-42, 1990.
[12] Y. Pantazis, Y. Stylianou and E. Klabbers, E. (2005). *Discontinuity Detection in Concatenated Speech Synthesis Based on Nonlinear Speech Analysis*. In Proc. of Interspeech, 2005.
[13] A.J. Viterbi (1967). *Error bounds for convolutional codes and an asymptotically optimal decoding algorithm*. IEEE Transactions on Information Theory IT-13, 260-269.
[14] G.D. Forney (1973). *The viterbi algorithm*. Proceedings of the IEEE 61, 268-278.
[15] T. Dutoit (2004). *TTSBOX 1.0: A Matlab toolbox for teaching Text-TO-Speech Synthesis*. Faculté polytechnique de Mons, 2004.
[16] T. Dutoit and M. Cernăk (2005). *TTSBOX : A Matlab toolbox for teaching Text-To-Speech Synthesis*. IEEE-ICASSP, 2005.
[17] S.F. Chen and J. Goodman (1998). *An empirical study of smoothing techniques for language modeling*. Center for Research in Computing Technology, Harvard University, Cambridge, Massachusetts, 1998.