Towards Fully Automatic Speech Processing Techniques for Interactive Voice Servers

Gérard Chollet[1], Jan Černocký[2], Guillaume Gravier[1], Jean Hennebert[3,4], Dijana Petrovska-Delacrétaz[3], and François Yvon[1]

[1] ENST - CNRS URA 820, 46 rue Barrault,
75634 Paris Cedex 13, France
{chollet, gravier}@tsi.enst.fr, yvon@inf.enst.fr
http://www.enst.fr/
[2] Institute of Radioelectronics
Technical University Brno, Czech Republic
cernocky@urel.fee.vutbr.cz
[3] Circuits and Systems Group
Swiss Federal Institute of Technology,
1015 Lausanne Switzerland
dijana.petrovska@epfl.ch, jean.hennebert@ubs.com
http://circwww.epfl.ch
[4] Ubilab, UBS IT Innovation Laboratory
Bahnhofstrasse 45 P.O. Box CH-8098 Zurich, Switzerland

Abstract. Automatic Speech Processing (Speech Recognition, Coding, Synthesis, Language Identification, Speaker Verification, Interpreting Telephony, etc.) has progressed to a level which allows its integration in the context of Interactive Voice Servers (IVS). The description of a personal telephone attendant ('Majordome') focuses on some of the issues in the development of IVS. In particular, users should be allowed to dialogue with automatic systems over the telephone in their native language. To achieve this goal, we propose an approach called ALISP (Automatic Language Independent Speech Processing). The needs for ALISP are justified and some of the corresponding tools are described. Applications to very low bit-rate coders, automatic speech recognition and speaker verification illustrate our proposal.

1 Introduction

An increasing amount of interpersonal communication is realized over the telephone. The widespread use of mobile telephones accentuates this situation. In many occasions, a telephone call is either quite disturbing or does not reach the desired person. Voice messaging systems, either centralized or individual, provide a partial but often frustrating solution. Recent progress in *Automatic Speech Recognition, Understanding and Synthesis* creates new opportunities for a profitable speech market. Many products are available for call centers, automatic telephone attendants, information and reservation systems, and many more are under field tests. They are grouped here under the denomination of

Speech Processing, Recognition and Artificial Neural Networks: Proceedings of the 3rd International School on Neural Nets "Eduardo R. Caianiello"

telephone *Interactive Voice Servers* (IVS). Such servers interact with the caller using speech input (recognition and understanding) and output (synthesis). For some applications (telephone card, banking, etc.), the identity of the caller is of interest and *Speaker Recognition* technology is deployed. The description of a personal information server ('Majordome') illustrates here many of the desired features of an IVS. The 'Majordome' serves as a telephone attendant which identifies familiar voices and verifies the identity of authorized users. It also recognizes proper names and spellings and can be used to access e-mail, fax-mail, voice-mail and web pages from any telephone.

Although existing servers often restrict the language, the words and the syntax they can interpret, the future calls for 'Unrestricted Vocabulary Continuous Speech Recognition' for any speaker, any language, any dialect and sometimes under noisy or distorted conditions. State-of-the-art large vocabulary continuous speech recognition technology relies on stochastic models of a limited set of acoustic units such as phones (the acoustic realization of phonemes). The estimation of the parameters of these models requires the availability of large phonetically annotated speech databases. The annotation and labeling of these databases is time consuming (and therefore expensive) and prone to errors. A different approach is developed here: *Automatic Language Independent Speech Processing* (ALISP) tools are proposed as automatic learning techniques to solve some speech processing problems when *no* labeled data are available. In particular, Speech Recognition, Speaker Verification and Language Identification are possible within this framework. ALISP tools can be used to define a set of universal acoustic units without any phonetic knowledge. Large speech corpora could be used in this framework with no requirements for phonetic annotation nor labeling. It is argued that the development of a very low bit-rate speech coder permits an evaluation of segmental models and a potential generalization to all languages of the world. Variable length sequence modeling (also refered to as 'multigrams') is one of the generic ALISP tools which finds applications at different levels of speech processing. It is applied here at the acoustic and lexical levels but could potentially be used for language modeling and translation.

This chapter is organized as follows: the 'Majordome' is first described to illustrate some of the problems of Interactive Voice Servers which motivate our emphasis on ALISP tools for very low bit-rate vocoding and lexical encoding, on the phonetization and recognition of proper names and spellings, and on speaker verification. Results are given concerning our experiments using some of the ALISP tools for the NIST'98[1] speaker verification evaluation campaigns.

[1] NIST organizes every year an evaluation of speaker verification systems (`http://www.nist.gov/speech`). A unique data-set and evaluation protocol are provided to each participating laboratory, so that intra- and inter-laboratory algorithms comparisons are significantly easier.

2 Interactive Voice Servers

An automatic system connected to the telephone network and able to manage some vocal dialogue with a caller will be denominated an Interactive Voice Server (IVS) in the context of this paper. Automatic train and airline travel information and reservation, stock quotes [23] or automatic telephone assistance systems [17] are typical examples which require different levels of complexity in speech recognition and synthesis. A telebanking system will also necessitate some form of identity verification, speech being the prefered support in this context.

The size and diversity of the population that will use the server, the restrictions on the dialogue, the size of the lexicon of interest, the necessity of performing caller identification and/or verification are all features which influence current research and development for IVS. Our 'Majordome', a personal information server, will offer many of the following features; it

- accepts any calls (the potential population is very large) in any language and dialect,
- recognizes proper names and spellings,
- interprets messages in order to summarize or translate them,
- identifies familiar callers (open-set speaker identification),
- adapts to the voice of the caller,
- verifies the identity of clients from the pronunciation of their name, password (text-dependent speaker verification) and continuously during the dialogue (text-independent),
- browses the web to satisfy any request from the caller (the application domain may not be restricted).

Let us first give some motivations for such a 'Majordome', then indications about existing hardware and software, an example on how it could be used and implications concerning speech technology.

2.1 Motivations for a 'Majordome'

Time and space asynchronous personal communication is achieved by various means: surface, electronic, voice and fax mail. However, those means are not equivalent in their usefulness. For example, surface mail can be used to transmit nearly any type of objects (letters, books, audio tapes, photographs, etc.), but takes a long time to reach the recipient. On the other hand, electronic, voice and fax mail are delivered almost instantaneously. Voice mails, faxes and e-mails have different areas of use. While it is easy to transmit some pages of source code via e-mail or even fax[2], or to transfer a file using e-mail, it is not convenient to transfer a voice message by fax or to give much details about an image or a drawing using speech. But, as versatile as fax and specially electronic mail may be, there is still a problem accessing the information. Not everybody owns a

[2] Although Optical Character Recognition or retyping is necessary in order to use the code and not just to read it

Personal Digital Assistant capable of connecting to one's mailbox via a cellular phone. On the other hand, it just takes a simple (public) phone to access an answering machine from any location in the world and therefore be up to date about the latest calls. Hence, came the thought of developing an '*intelligent answering machine*', that is not only capable of storing voice messages and, faxes and e-mails, but also interprets them so that the owner of these messages could access them from any telephone upon verification of his identity. Some interests in the 'Majordome' project are, amongst others:

- a telephone attendant when the owner is absent or too busy to answer the phone,
- a transfer of urgent calls,
- a voice controlled interface, i.e. no more nestling around with telephone pads,
- using speaker verification to restrict access to the accounts,
- integrate text-to-speech technology for reading faxes and e-mails to the accounts' owner,
- Integrated Services Digital Network (ISDN) based, computer-sided interface to the telephone lines,
- using Optical Character Recognition and handwriting recognition to determine the fax recipient and sender names and interpret the content of the message,
- the possibility of dictating an e-mail, fax or voice message to be delivered by the Majordome,
- the owner could ask any question about any subject. The Majordome will browse the web to find some relevant answers.

Furthermore, it is thought of developing a client software that permits access to Majordome on a HTTP/HTML basis, so that the information can be retrieved from an Internet account as well.

2.2 Hardware and Software

Some virtual assistant services (Wildfire[3], Portico[4]) are being commercialized. They use proprietary hardware and software to be shared by multiple users. A more personal solution is proposed here: a PC with an ISDN board is the minimum hardware necessary to install a 'Majordome'. ISDN provides the proper telephone interface to handle simultaneously voice and fax calls. Since every standard ISDN board is able to handle two channels (any combination of two incoming or outgoing calls) at the same time, different numbers are given to the different services, i.e. one number for incoming calls, one for faxes and a third one for communications with registered users.

Another advantage of ISDN boards is the fact that multiple applications can access to them, so that the Majordome server will not prevent other applications from running on the same computer, like dial-up networking or Internet access.

[3] `http://www/mrtramp.com/Wildfire.htm`
[4] `http://www.generalmagic.com`

The ISDN board is programmed using Microsoft Visual C++ 5.0 Professional and the Common ISDN Application Programming Interface (API), a now widely accepted and operating system independent standard for developing applications for ISDN boards. The problem is, of course, that this limits Majordome to some lowest common denominator, i.e. that features like call redirection might not be accessible using this kind of API. Anyway, this approach is far better than writing a program for a specific card and thus limiting the possible equipment on the target server. No decision has yet been made about the language that is going to be used for the client software, since it may be a very convenient, time and cost saving way to write this client software using Java, but some requirements for the client are not yet met by the Java language, such as operating system independent audio recording.

2.3 Some Examples of Use of Majordome

Let us now consider a case where the Majordome is centralized at the level of a company. This company owns a digital Public Access Branch eXchange (PABX) telephone system capable of transferring calls in case of no response. It sets up the Majordome server properly on a PC, assigning it the phone number #01 for incoming phone calls, #02 for incoming fax messages and #03 for communications with users that have an account on that Majordome server. In addition to that, the PC is connected to the company wide intranet. Let's assume person A calls person Z, who has an account on the Majordome. If Z is away from his phone or does not want to answer, the PABX of the company transfers the call to #01 after 5 tones. The Majordome picks up the call, A is asked by Majordome to give his name and the name of the recipient. The names are tentatively recognized and in case of ambiguity, spelling is requested. This information is used to determine the correct account by speech recognition and to inform Z that he received a call from A. If Z has left a phone number to Majordome, the Majordome attempts to reach him on line #03. Otherwise, Majordome knows from the Intranet whether or not Z is connected on a terminal and sends a warning on that terminal for connection on line #03. Z has therefore the possibility of monitoring the dialogue between A and the Majordome. In the mean time, Majordome asks A to deposit a message. Z can take up the call at any time. If he chooses not to do so, the recorded message is placed into Z's mail box.

Simultaneously, B could send a fax to the Majordome server, thus dialing #02. Majordome attempts to find out the sender and recipient names of the fax using OCR and handwriting recognition. If it fails to recognize the recipient name, it transfers the fax to an operator. If it recognizes Z as the recipient, it stores that fax into Z's mail box and try to contact Z on the intranet.

If Z is outside the company, he may want to call Majordome at any time to access his mail box or get some answers about any question of interest to him. He calls Majordome on line #03. He is asked to give his name and his password. He is identified by the name and verified by the pronunciation of both the name and the password. When Z passes both test, Majordome tells him that he received a voice message from A, a number of e-mails and a fax. He

can ask Majordome to read a summary of the messages to him, to read just the first n lines or the subject, to delete the messages or to forward any of them to someone else. If Z can not understand the name of the sender, Z can also ask Majordome to spell the name. Z may ask Majordome to attempt to interpret the fax content. He could ask Majordome to forward that fax to a given number. Z has the possibility of making vocal database inquiry through his Majordome. In particular, Majordome may browse for information on the Intranet and the Internet upon request.

2.4 Interactive Voice Servers and ALISP

The success of Interactive Voice Servers may depend on the ergonomy and robustness of the dialogue. A caller should be able to use his native language and therefore, specific recognizers and synthesizers must be developed for as many languages and dialects of the world for which a market exists. The next section proposes an approach to Automatic Language Independent Speech Processing (ALISP) which may facilitate such developments both at the acoustic and linguistic levels. No restriction on the vocabulary or the syntax should be imposed to a caller. Proper names recognition is of crucial importance for a personal information or directory assistance server. Spelling could be used if necessary to achieve a sufficient level of accuracy. Sect. 4 of this chapter deals with the recognition of proper names and spellings. The identification of a caller may be necessary to restrict access to personal information. This could be done explicitly by requesting a name and password and implicitly from the speech signal produced by the caller.

3 Automatic Language Independent Speech Processing

Automatic Language Independent Speech Processing (ALISP) adapts and applies Machine Learning algorithms to the Speech and Natural Language fields. It is assumed that an automaton can learn from examples. Children acquire a language from interactions with other children and adults. They do not need an explicit labeling of the data they receive. In a similar way, ALISP should discover the structure of speech and natural languages from large corpora of speech signal and texts.

Speech is a continuous signal to which some form of symbolic representation must be associated. Lexical units (words) seem to be a useful level common to speech and natural language processing. Words can be described in terms of smaller units. Linguists have proposed the phoneme as the formal unit to distinguish a 'minimal pair' of words (the pronunciation of the English words 'tee', 'pea', 'key', 'bee', 'me', 'fee', 'see', 'we' differs in their initial part). The problem is that phonemes in different contexts exhibit different acoustic characteristics. We propose to find a set of segmental units automatically from recordings of continuous speech. These units are evaluated in the context of a very low bit-rate coder (see Sect. 3.4).

Variable length sequence modeling is a general tool to discover regularities in strings of symbols. We first introduce this technique and suggest its application at different levels of speech and language processing.

3.1 Variable Length Models of Language Processing

Most application-oriented models of language processing rely upon a common representation of linguistic data, usually taking the form of sequences of primitive discrete symbolic units. Syntactical analysis decomposes sequences of words into hierarchical sequences of syntagmatic categories, morphological analysis decomposes sequences of phonemes or letters (word forms) into sequences of morphemes, etc. These (minimal) units are assumed to be provided by traditional linguistic descriptions.

The multigram model [5] promotes quite a different view: the segmentation units should also be subject to some kind of discovery procedure, in order to model more accurately the facts that i) relevant (or optimal) units for a given task might cover a variable number of "primitive" units, and that ii) dependences between adjacent units might span over a variable length number of "primitive" units. Grapheme-to-phoneme conversion is a clear-cut example of i): many groups of letters in fact function as a whole, like *ph*, *sh*, etc.; similarly, the modeling of co-articulation effects in speech recognition or in concatenative speech synthesis is a well-known case of variable-length dependency, when expressed at the phonetic level of representation.

This model has been found suitable for a wide range of applications, like the identification of multi-word units in statistical language models [12], the specification of a minimal set of units for speech synthesis [4], automatic segmentation of texts [5], etc.. In this section, we briefly survey two applications of this model which are of particular interest for building interactive voice servers, i.e. the identification of recognition units (see Sect. 3.2), and the construction of proper names pronunciation dictionaries (see Sect. 4).

3.2 Automatically Derived Sub-Word Units

Sub-word units are widely used in various domains of speech processing. Classically, they are based on phonemes or phoneme-related units such as context-dependent phonemes, syllables, etc. Their search requires an important amount of phonetic and linguistic knowledge. In order to train a speech processing system, annotated training databases are necessary. The annotation using phonetically-derived units is a time-consuming, costly and error-prone task. Even if natural language processing can not be done without phonetic and/or linguistic expertise, recent advances in Automatic Language Independent Speech Processing [8] have shown, that many tasks relying currently on such knowledge can be performed using data-driven approaches. From a practical point of view, extensive human efforts can be replaced by an automated process. This fact could bring revolutionary changes to the methodology of speech processing.

3.3 ALISP tools

Several tools are used for unsupervised search of acoustically coherent speech units. They are based on *speech signal data* rather than on the textual representation of the latter. The tools are modular and Fig. 1 gives an example of how they are linked in the framework of speech coding.

As a first step, the goal of *temporal decomposition* is to detect quasi-stationary parts in the parametric representation of speech. This method, introduced by Atal [1] approximates the trajectories of parameters $x_i(n)$ by a sum of m *targets* a_{ik} weighted by *interpolation functions*

$$\hat{x}_i(n) = \sum_{k=1}^{m} a_{ik}\phi_k(n), \quad \text{or} \quad \underset{(P\times N)}{\hat{\mathbf{X}}} = \underset{(P\times m)}{\mathbf{A}}\ \underset{(m\times N)}{\mathbf{\Phi}}\ , \tag{1}$$

in matrix notation, where the lower line indicates matrix dimensions. The initial interpolation functions are found using local Singular Value Decomposition with adaptive windowing [3], followed by post-processing (smoothing, decorrelation and normalization). Target vectors are then computed by $\mathbf{A} = \mathbf{X}\mathbf{\Phi}^{\#}$, where $\mathbf{\Phi}^{\#}$ denotes the pseudo-inverse of the matrix. Interpolation functions and targets are iteratively locally refined by minimizing the distance between $\mathbf{X}$ and $\hat{\mathbf{X}}$. Intersections of interpolation functions define speech segments.

Unsupervised clustering assigns segments to classes. Vector quantization (VQ) is used for automatic determination of classes. A K-means algorithm with binary splitting is used to train the VQ codebook $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_L\}$. Training is performed using vectors positioned at the gravity centers of the interpolation functions, while the quantization takes into account entire segments using cumulated distances between all vectors of a segment and a code-vector. Temporal decomposition along with vector quantization can produce a phone-like segmentation of speech.

Multigrams [11] may serve for finding characteristic sequences of quantized events or of segments determined by Hidden Markov Models (HMMs). The method is based on finding the optimal segmentation of a symbol string into variable length sequences called multigrams, using a maximum likelihood criterion:

$$\mathbf{X}^{\star} = \arg\max_{\forall X} \mathcal{L}(\mathbf{O}, \mathbf{X}|\{x_i\}) \ , \tag{2}$$

where $\mathbf{O}$ is the string of observations, $\mathbf{X}$ is the segmentation and $\{x_i\}$ is the codebook of available multigrams. The likelihood is given by the product of probabilities $\mathcal{P}(x_i)$ of multigrams in the segmentation $\mathbf{X}$. These are not known and must be estimated on the training corpus using iterations of segmentation according to (2) and of probabilities re-estimation using sequence counts.

Finally, HMMs can be used to model the units. HMM parameters are initialized using context-free and context-dependent Baum-Welch training with Temporal Decomposition + Vector Quantization or Temporal Decomposition + Vector Quantization + Multigrams transcriptions, and refined in successive steps of corpus segmentation using HMMs and model parameters re-estimation. The

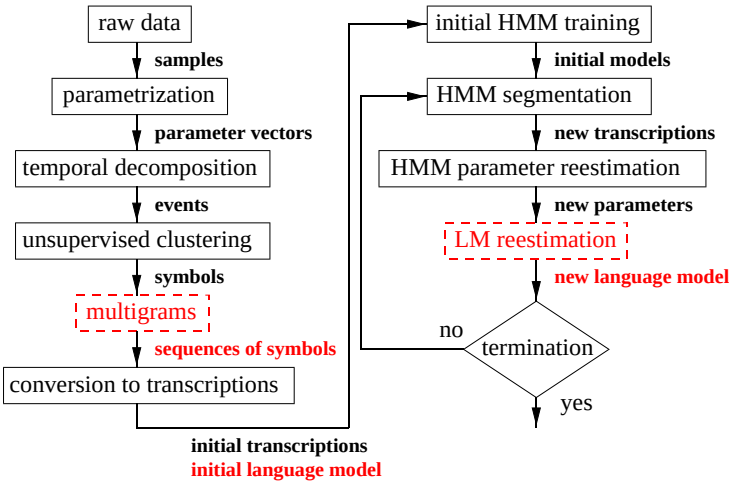

Fig. 1. Data-driven derivation of coding unit set in VLBR phonetic vocoder

speech represented by the observation vector string $\mathbf{O}$ can then be aligned with models by maximizing the likelihood

$$\arg\max_{\{M_1^N\}} \mathcal{L}(M_1^N|\mathbf{O}), \quad \text{where} \quad \mathcal{L}(M_1^N|\mathbf{O}) = \frac{\mathcal{L}(\mathbf{O}|M_1^N)\mathcal{L}(M_1^N)}{\mathcal{L}(\mathbf{O})} \ . \tag{3}$$

M_1^N is the sequence of models and $\mathcal{L}(M_1^N)$ the prior probability of M_1^N determined by a language model (LM).

3.4 Very Low Bit-Rate Coding

Very low bit-rate (VLBR) coding with data-driven units is a framework to test the efficiency and usefulness of the ALISP approach. In this area, the task of pronunciation modeling does not need to be resolved, but the efficiency of the algorithms is evaluated by re-synthesizing the speech and by comparing it to the original. If this output is intelligible, one must admit, that this representation is capable of capturing the acoustic-phonetic structure of the message and that it is appropriate also in other domains. Moreover (in contrast with a classical approach, where the unit set is fixed a-priori and can not be altered), the coding rate in bps and the dictionary size carry information about the efficiency of the representation, while the output speech quality is related to its accuracy.

The flow-chart given in Fig. 1 shows how data-driven derived coding units are obtained using a training corpus. With these units, the test corpus is encoded by aligning the data with HMMs and the efficiency of coding is evaluated by the average bit-rate R_u (in bps), assuming uniform encoding of sequence indices. Prosody information is not taken into account and synthesis is done using representatives drawn from the training corpus. Experimental setup and results

Table 1. Summary of VLBR coding experiments. TD stands for temporal decomposition, MG for multigrams, CU for coding units and R_u for the average bit-rate

database	PolyVar	BU Radio Speech Corpus
language	Swiss French	American English
speakers	1 (the most represented)	2 (F2B, M2B)
parameterization	10 LPCC,ΔLPCC,E,ΔE	16 LPCC,ΔLPCC,E,ΔE
TD	avg. 15 events/sec	avg. 17 events/sec
VQ codebook	64	64
MGs prior to HMMs	yes	no
HMMs to train	1666	64
HMM refinements	1	5
MGs after HMMs	no	yes
coding units (CU)	1514	722 (F2B), 972 (M2B)
representatives per CU	8	8
R_u [bps] (test set)	120	110 (F2B), 119 (M2B)

are summarized in Table 1. In the first case, the synthesis was done by a simple concatenation of representative signals. In Boston University (BU) experiments, the synthesis was Linear Predictive Coefficients-based (LPC) using the original prosody. In both sets of experiments, the resulting speech was found intelligible, but the quality is significantly worse than for codecs at several kbps. Details and speech files can be found in [40] and its related Web-page.

3.5 Comparison with Phonetic Alignments

The phonetic alignments available with the BU corpus allowed us to investigate the correspondence of phones and ALISP units. These alignments were given by BU using a segmental HMM recognizer constrained by the possible pronunciations of the utterances [31]. In our comparison, the alignment files without hand-corrections were used. Phonetic alignments were taken as reference and ALISP segmentations (last generation HMM) were compared against them. The measure of correspondence was the relative overlap r of an ALISP unit with a phoneme. The results are summarized in a confusion matrix $\mathbf{X}$ $(n_p \times n_a)$, whose elements are defined as follows:

$$x_{i,j} = \frac{\sum_{k=1}^{c(p_i)} r(p_{i_k}, a_j)}{c(p_i)} \quad , \tag{4}$$

where n_p and n_a are respectively the sizes of phoneme and the ALISP unit dictionaries, p_i is the i-th phoneme, a_j is the j-th ALISP unit, $c(p_i)$ is the count of p_i in the corpus and $r(p_{i_k}, a_j)$ is the relative overlap of k-th occurrence of p_i with an ALISP unit a_j. The columns of $\mathbf{X}$ are rearranged to let the matrix have a quasi-diagonal form[5] and the resulting matrix is given in Fig. 2. On contrary

[5] Thanks to Vladimír Šebesta and Richard Menšík (Inst. of Radioelectronics TU Brno) for their help in the visualization of confusion matrices.

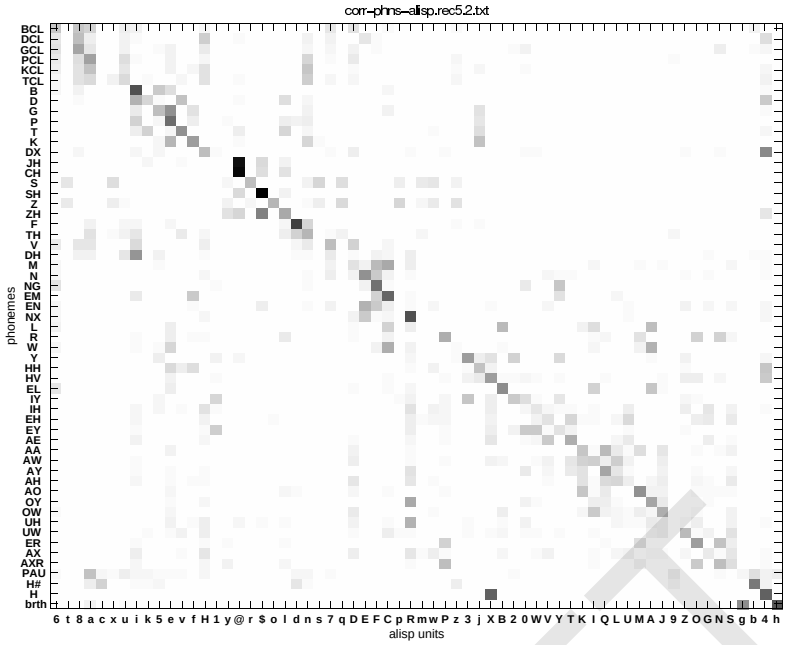

Fig. 2. Correspondence of ALISP segmentation and phonetic alignment for speaker F2B in BU corpus. White color corresponds to zero correlation, black to maximum value $x_{i,j}$=0.806

to BU alignments, where stressed vowels are differentiated from unstressed ones, we used the original TIMIT phoneme set.

Although these experiments showed a correlation of phonemes and ALISP units, an ALISP recognition system should not be based on direct phoneme–ALISP mapping. It would be more efficient to represent the target dictionary as probabilistic combinations of sequences of ALISP units. The work of Fukada [16] on phoneme and word based automatically derived segment unit composition, and Deligne's joint multigrams [11] did bring interesting insights on this representation.

4 Recognition of Proper Names and Spelling

4.1 Introduction

Automatic retrieval of names from their pronunciation and spelling is a popular topic in speech recognition. It is also a key problem for IVS since many aspects of a system like Majordome (see Sect. 2) critically rely on its ability to handle proper names in a right way. For instance, the identification procedure requires the recognition of the owner account's name; mail reading requires the ability to utter accurately (intelligibly) the sender's name, etc.

Many studies address the problem of alphabet recognition, which is known to be a difficult task because of acoustic similarities between some letters (e.g. the well-known E-set in English). When working with telephone quality speech, the confusability between letters increases drastically. For example, Cole *et al.* presented experiments with telephone speech for the recognition of English and French alphabets [9] and [36]. Letters, separated by pauses, are segmented and then classified using a Multi-Layer Perceptron (MLP) with phonetic features. More recently, Hidden Markov Model (HMM) based methods have been proposed, as in [21] and [22] where letter models are used. The first study compares dynamic time warping (DTW) and HMM based lexical search strategies to retrieve names from natural spelling. HMM based lexical search can also be seen as a DTW whose insertion, deletion and substitution costs are learnt from the training data. The second study is based on a multi-level classification with a N-best approach, using DTW and a restricted grammar for the lexical search. These studies however fail to address a number of problems that are critical in the perspective of real applications. Firstly, they greatly underestimate the variability observed in real-life spellings. Secondly, phonetic knowledge is mainly used to improve letter discrimination but not for the lexical access. Finally, the name itself (i.e. the pronunciation of the name) is a source of information for name retrieval applications that has rarely been used (see however [27]).

In this section we present a system for the recognition of proper names from the pronunciation and spelling. The results reported here are described in more details in [18] and [28]. Sect. 4.2 presents an overview of the system where the two main stages are described. The first stage consists in acoustic decoding while the second one consists in lexical search. The automatic generation of the lexicon from orthographic names is also explained. Results of several optimization experiments are presented in Sect. 4.3. It must be stressed that currently, this system does not use any of the ALISP techniques proposed so far in the paper. However, we explain in Sect. 4.4 how multigrams can be extended to induce the pronunciation of proper names.

4.2 System Description

The name recognition system presented here is divided in two successive stages. In the first stage, acoustic decoding, based on HMM, is performed without any knowledge of the lexicon content. In the second stage, the recognized sequence of phones and letters is matched against all the entries of the lexicon to find out the name. Furthermore, this system is designed to be as extensible as possible and therefore the lexicon is generated automatically and the acoustic models are trained in an unsupervised manner. The advantage of the two-pass architecture compared to a Large Vocabulary Continuous Speech Recognition (LVCSR) based system, where the decoding is constrained by the lexicon, is that the former is not limited by the size of the lexicon and can therefore deal with larger lexicons.

Lexicon. An entry in the lexicon contains one or several phonetic transcription(s) of the name as well as one or several possible spellings. The entries are

generated automatically from the orthographic transcription of the name using a grapheme to phoneme converter for the pronunciation(s) and a rule-based system for the spelling(s). The grapheme to phoneme converter used here was developed during the course of the Onomastica project [37], [43] and has been explicitly devised to cope with proper names idiosyncrasies. It also contains a module for recognition of proper name origins and is likely to output pronunciation variants. The possible spellings of a name are generated using a rule-based system which considers all the possible pronunciations for each cluster of letters. For example, the letter "é" can be spelled "*e accent aigu*", "*e aigu*" or "*é*"; the cluster "nn" can be spelled "*deux n*" or "*n n*" etc.

In the remainder, the i-th lexicon entry, denoted e_i, will be refered to as $\{n_{i,j=1,...,N_i}, s_{i,j=1,...,S_i}\}$, where $n_{i,j}$ (resp. $s_{i,j}$) is the j-th pronunciation (resp. spelling) variant.

Acoustic modeling. Phone models are adequate to recognize the pronunciations while letter models are better adapted to spelling recognition. It must be stated that the word "letter" here designs the alphabet letters plus some additional words used for spellings in French (such as "accent", "trait", "d'union", etc.) which are of course also modeled. The acoustic modeling relies on Hidden Markov Models. Phone HMM parameters are estimated on a training corpus from the speech data and the orthographic transcription. First, the training corpus is automatically segmented from its orthographic transcription, using task-independent phoneme models, trained on sentences of the Swiss-French Polyphone database[6] [7], (also used as bootstrap models). The parameters of the bootstrap models are then re-estimated on the pronunciations. Letter models are created by concatenating the bootstrap models corresponding to the letter pronunciation, and then performing re-estimation. Because of the possible liaisons after some words such as "*deux*", some letters may have two models, one with the liaison and one without.

The grammar used for the acoustic decoding is rather simple. A pronunciation can be any, possibly empty, sequence of phones while a spelling is any, possibly empty, sequence of letters. Optional silences can be found at the beginning and at the end of an utterance, and between the pronunciation and the spelling. A short silence (i.e. a silence model with a skip transition) is forced after each letter since previous studies [33] showed that this technique significantly improves the accuracy of the letter recognition. The optimization of the system parameters on the training corpus is presented in Sect. 4.3.

The speech signal is encoded using 12 Mel frequency cepstral coefficients and log energy, with first and second order derivatives, computed every 10 ms on 25.6 ms frames.

[6] Distributed by ELRA, `http://www.icp.inpg.fr/ELRA`

Name retrieval strategy. The distance between a lexical entry e_i and the form $r = (r_n, r_s)$ recognized during the first stage is defined by:

$$D(e_i, r) = \beta \min_j d(r_s, s_{i,j}) + (1 - \beta) \min_j d(r_n, n_{i,j}) \ ,$$

where r_s (resp. r_n) is the recognized spelling (resp. pronunciation). The dissimilarity measure $d(.,.)$ is computed by dynamic alignment using specific costs for each possible substitution, insertion and deletion. Those costs are usually arbitrarily fixed. However, as some pairs of symbols (letters or phonemes) are more confusable than others, it seems natural to assign a smaller cost for the substitution of confusable symbols. Therefore, the weighted cost for the substitution of x by y is $-\log(p(y|x))$, where $p(y|x)$ is estimated using the confusion matrix on the training corpus. The same procedure is used to determine insertion and deletion costs. This approach is somewhat similar to the HMM based alignment procedure used in [21]. Instead of using phonetic knowledge to achieve a better discrimination as in [24], this knowledge is learnt from the training data and used for name retrieval rather than for symbol recognition. Parameter β allows to balance the respective contributions of the spelling and pronunciation.

4.3 Experiments

Database. Experiments are carried out on the Swiss-French Polyphone database which was recorded over the telephone with 5,000 speakers calling once, a call including the pronunciation and spelling of 3 names. Spellings were prompted but no specific spelling guidelines were given to the callers. As a result, in addition to "standard" spellings, the corpus contains occurrences of comparisons, such as "*a comme alain*" (a not so uncommon way to minimize confusions between letters), occurrences of aeronautic-like spellings, eg. "*alpha bravo ...*", etc.

A subset of the database is used, containing 11,920 speech segments from 3,998 speakers. This subset was divided in three corpora. The first one is the train corpus, containing 5,390 segments from 3,223 speakers. The remaining items belong to the test corpus, from which a special subset, the "*clean*" test corpus, is extracted. This "*clean*" test corpus contains the items for which the spelling conform to the "standard" spelling conventions. The "*clean*" test corpus contains 5,015 segments from 3,097 speakers, corresponding to 3,478 different names and is used to evaluate the quality of the acoustic modeling. The entire corpus contains 8,261 names.

Acoustic decoding. The acoustic decoding without language model or fixed transition penalty, gives very poor results, specially at the phone level. Indeed for the phones we have an accuracy of 14.4% on the training corpus which drastically decreases to -6.7% on the clean test corpus. The recognition of letters is much more reliable since the accuracy is 69.7% on the clean test corpus. So weak performances are due to the huge amount of insertions. In order to reduce the number of insertions, the fixed transition log-probability p was introduced.

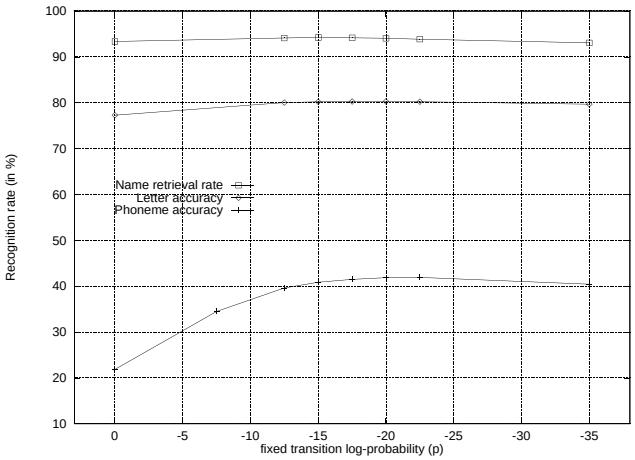

Fig. 3. Decoding accuracy on the train corpus as a function of the fixed transition log-probability p

Table 2. Recognition rate accuracy (in %) for phones and letters

	phones		letters	
	$p = -17.5$	$s = 8$	$p = -17.5$	$s = 8$
train	54.3/41.5	59.3/51.5	83.4/80.3	89.7/84.1
clean test	56.2/34.4	60.2/47.5	82.7/78.4	88.0/81.9

Fig. 3 plots the correct recognition rates and the accuracy as a function of the fixed probability p. The name retrieval rate is also plotted and it can be seen that, though the phone accuracy significantly increases, the name recognition rate does not really improve. This is explained by the fact that the costs for the dynamic alignment of two forms are learned on the training corpus. It also points out that the technique which consists in determining the substitution, insertion and deletion costs is effective. The use of a bigram language model instead of a fixed transition probability was also tested and results are reported in Fig. 4 where the phone and letter accuracy are reported for several values of the fudge factor s. Better accuracies are obtained with the language model than with the fixed probability but no real difference is observed at the name recognition level. Table 2 gives the recognition rates and the accuracies for the optimal values of s and p on the training and clean test corpora. In each cell of the table, the left figure corresponds to the recognition rate while the right one corresponds to the accuracy. As can be seen, the language model significantly improves the quality of the phone decoding but letter recognition still remains more reliable.

A more detailed study of the letter recognition errors outlines the standard confusions between acoustically similar letters such as “b” and “d”, “f” and “s” or “p” and “t”. As noted in many studies on alphabet recognition (eg. [21] and

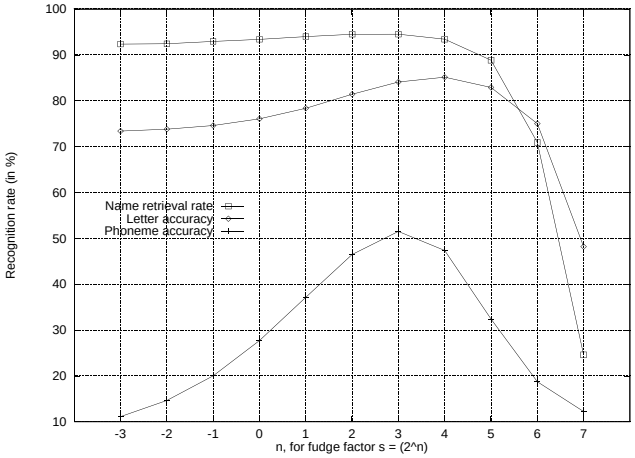

Fig. 4. Decoding accuracy on the train corpus as a function of the fudge factor

[24]), letter HMMs are not really able to distinguish two letters whose spellings only differ by a short transitional acoustic event. For example, letters "m" and "n" share the same vowel and are separable by the last consonant. But phonemes /*m*/ and /*n*/ are quite similar, and their confusion is also one of the most common substitutions at the phoneme level.

Name retrieval. Results on the name recognition rates are reported. To measure the respective importance of the pronunciation and the spelling, the recognition rate is computed on the training corpus for various values of β. Results are reported in Fig. 5 for $p = -17.5$. An optimal value is found for $\beta = 0.6$ which reflects the fact that the pronunciation is a valuable source of information for the task. Similar curves are obtained on the clean and complete test corpora with recognition rates of respectively 84.0% and 79.1% at the optimal point. When using a language model instead of the fixed transition log-probability, the recognition rates are slightly better. The effectiveness of the learnt dynamic alignment cost is clearly shown since for a recognition rate of 56.2% on the clean test corpus for binary costs, the rate increases to 82.1% with learnt cost.

An exhaustive search strategy across the entire lexicon is rather time consuming. Therefore, a fast pre-selection, based on the spellings of a set of possible entries in the lexicon was developed. This approximative search is based on a variant of the algorithm originally proposed in [29] for error-tolerant lexical recognition. For a 8K lexicon, the search is about 9 times faster with a preselection of 100 lexical entries with a very small decrease of the performances (83.2% instead of 83.6%).

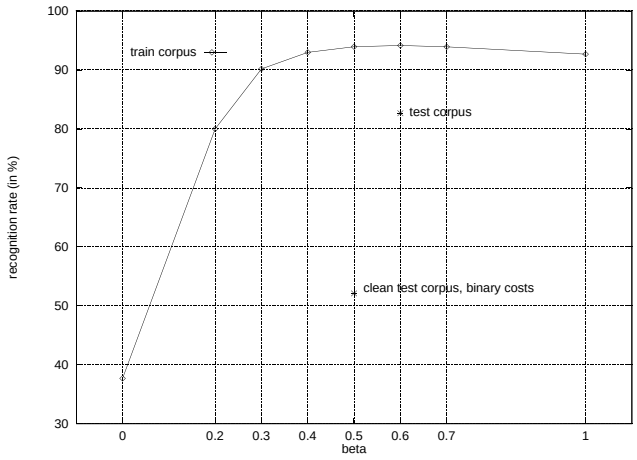

Fig. 5. Name recognition rate as a function of β

4.4 Inducing the Pronunciation of Proper Names

Independent of the speech recognition and synthesis methods used, a representation of the proper name pronunciation is necessary, which is not easily obtained [37]. In fact, proper names represent a challenging task for traditional, rule-based transcription systems: they often contain very unusual letter-sound associations, a fact which is dramatically severed by the variety of linguistic origins of names [41]. Given the very high number of different proper names, and the pace of apparition of new items, pure lexical approaches, only providing a limited coverage of the proper name diversity, are also bound to fail.

The methodology we advocate consists in using self-learning techniques allowing to generalize over existing pronunciation dictionaries. Several well known learning algorithms have been proposed and used to this ends: neural networks [38], decision-trees [14], nearest neighbors [39], etc. However, these techniques make assumptions regarding phonetic representations, in particular that phonetic and graphemic strings have approximately equal length, and that the former are shorter than the latter. Chunk-based transcription models [10], [25], [42], which dispense with this kind of assumptions, seem to fit in better with our general principles: language independence and use of acoustic or linguistic units for representing pronunciations. In line with the core ideas of ALISP, we develop hereafter another instance of chunk-based model, namely the joint-multigram model [13], which extends the multigram model (see Sect. 3.1) to the case of bi-dimensional streams.

The basic idea of the joint-multigram model is to automatically identify recurrent joint sequences in a pronunciation dictionary, and use them as the primary units for the transcription of unknown words. Formally, a joint sequence $\begin{bmatrix} a_1 \ldots a_n \\ \alpha_1 \ldots \alpha_m \end{bmatrix}$ is made of two parallel chunks, corresponding in our context respec-

tively to sequences of graphemes and of phonetic or acoustic units. In its simplest form, the joint-multigram model considers each pronunciation sample as the result of the concatenation of joint sequences, the borders of which are not known. Training simply consists in finding, in a set of examples, the most likely pairing of variable-length sequences of graphemes with variable-length sequences of phonetic/acoustic units. The resulting set of sequences, along with their probability of co-occurence, can be used to infer, through a sequence-by-sequence decoding process, the string of units Ω which best matches a given orthographic form O. This transduction task can be expressed as a standard maximum a posteriori decoding problem, consisting in finding the most likely $\widehat{\Omega}$ given O:

$$\widehat{\Omega} = \arg\max_{\Omega} \ \mathcal{L}(\Omega \mid O) = \arg\max_{\Omega} \ \mathcal{L}(O,\ \Omega) \ . \tag{5}$$

Under the traditional assumption that $L^* = (L^*_O, L^*_\Omega)$, the most likely joint segmentation of the two strings, accounts for most of the likelihood, the maximization step is rewritten:

$$\widehat{\Omega}^* = \arg\max_{\Omega} \ \mathcal{L}(O,\ \Omega,\ L^*_O,\ L^*_\Omega) \tag{6}$$

$$= \arg\max_{\Omega} \ \mathcal{L}(O,\ L^*_O \mid \Omega,\ L^*_\Omega)\ \mathcal{L}(\Omega,\ L^*_\Omega) \ , \tag{7}$$

by application of the Bayes rule. $\mathcal{L}(O,\ L^*_O \mid \Omega,\ L^*_\Omega)$ scores how well the graphemic sequences in the segmentation L^*_O match the inferred phonetic representation in L^*_Ω. It is computed as $\prod p(s_{(t)} \mid \sigma_{(t)})$, where the conditional probabilities are deduced from the probabilities $p(s_i, \sigma_j)$ estimated during the training phase. The term $\mathcal{L}(\Omega,\ L^*_\Omega)$ measures the likelihood of the inferred pronunciation: it can, for instance, be estimated as $\widetilde{\mathcal{L}}(\Omega,\ L^*_\Omega)$, using a language model. This decoding strategy is a way to impose syntagmatical constraints in the string Ω (here phonotactical constraints). The maximization (7) finally rewrites as:

$$\widetilde{\Omega}^* = Argmax_\Omega \ \mathcal{L}(O,\ L^*_O \mid \Omega,\ L^*_\Omega)\ \ \widetilde{\mathcal{L}}(\Omega,\ L^*_\Omega) \ . \tag{8}$$

This model, and extensions thereof [13], has been evaluated on a French lexicon of common words, and has proved to achieve satisfying results, both in terms of the identified segmentation, and in terms of the overall pronunciation accuracy. At current stage, however, its benefits are still limited by the need to learn the model parameters from transcriptions at the word level which are not directly obtained from raw speech data. The next step [8] is thus to extend this model and to enable training to take place directly on ALISP-units based transcription of spoken utterances.

5 Speaker Recognition

5.1 Introduction

The generic term of speaker recognition comprises all of the many different tasks of distinguishing people on the basis of their voices. There are *speaker identification* tasks which consist in telling who, among a set of possible candidates, pronounced the available test speech sequence. On the other hand, there are *speaker verification* tasks for which one must say whether a specified candidate pronounced the available test speech sequence or not. In this section, we focus on speaker verification, which is actually a decision problem between the two following classes: the *true speaker* (also denominated as *client*, *claimant* or *target speaker*) and the *other speakers* (usually noted as *impostors speakers*).

As far as the speech mode is concerned, speaker recognition systems are usually classified as text-dependent or text-independent. In *text-dependent* experiments, the text transcription of the speech sequence is known a priori, and is constrained to be the same for training and testing. The knowledge of what was said can be exploited to align the speech signal into discriminant classes (words or sub-word speech units). The main advantage is a fair recognition performance with small amount of speech signal needed for training and testing. Their major drawback is the poor security level (the system can easily be fooled using pre-recorded speech).

In *text-independent* tasks, enrollment and test speech are completely unconstrained. Such scenarios offer more flexibility and enable higher security against pre-recorded speech if random text-prompting is used. Nevertheless, as the foreknowledge of what the speaker said is not available, less precise models are generally used and larger quantities of speech signal are needed to achieve acceptable performances.

In between text-dependent and text-independent lie intermediate systems such as *customized-password* systems. In this case, enrollment speech is unconstrained since the user is prompted to chose himself one (or more) password while the test speech is constrained to be the same from session to session. This approach offers user-friendliness and relatively high security against recording attacks if more than one password is used. The accuracy is generally better than text-independent tasks since modeling can be more precise. Similar to the customized-password technique is the *knowledge-based* approach in which the system prompts the user for his name, birth-date, or other personal data. Again, the enrollment speech is not predictable while the test speech is the same from session to session.

5.2 Segmental Speaker Verification Based on ALISP Units

As speech recognition technology is developing fast, there is an increasing amount of opportunities for using speaker recognitions techniques. In this framework, text-dependent systems have limited potential applications, specially wherever

user convenience and security against pre-recorded speech is an issue. The flexibility of text-independent and customized-password approaches make them better candidates for direct applications into IVS, but their performances are not yet satisfactory for real applications. The reason is that current text-independent systems are usually based on modeling globally the probability density function (pdf) of the speaker feature vectors. Such global models have poor discriminant capabilities because the temporal information of the speech sequence is not taken into account and also because all the phonetic classes are represented using a unique model. One way to overcome this problem is to combine the text-independent approach with speech recognition. In such a way, the speech signal is segmented into sub-word classes (phonemes or other related speech units) and speaker modeling is performed more precisely for each category. Such systems are designed here as *segmental text-independent* systems to contrast with the usual global approach.

The segmental approach recovers some text-dependent advantages since the speech signal is aligned into classes but the implementation is different since we have no clue about what is said. Several studies, such as [15], [30], [32] and [19], have demonstrated that some phones are more speaker discriminant than others, suggesting that a different weighting of individual class decisions should be performed when computing the global decision. Two potential advantages can be pointed out: firstly, if the speech units are relevant, then speaker modeling is more precise, thus allowing better performances than the global approach; secondly, if speech units present different discriminative power, then better recombination of the decisions per class can be done. The disadvantage of this method is that accurate recognition of speech segments is required. Two alternatives are possible.

- The first possibility is to use Large Vocabulary Continuous Speech Recognition (LVCSR) systems that provide the hypothesized contents of the speech signal on which classic techniques can be applied. LVCSR uses previously trained phone models and a language model, generally a bigram or trigram stochastic grammar.
- The second possibility is to use Automatic Language Independent Speech Processing (ALISP) tools that provide a general framework for creating sets of acoustically coherent units with little or no supervision.

LVCSR systems although very promising for segmental approaches, require huge phonetically annotated databases, which are either costly or not available and are often dependent on the speech signal characteristics (language, speech quality, etc.). These arguments make them difficult to adapt to new tasks. On the other hand, ALISP offers an alternative when no annotated training data are available. These are the reasons that led us to investigate a text-independent segmental approach based on ALISP techniques. As detailed hereafter, we propose to use the temporal decomposition followed by vector quantization to automatically obtain classes of sounds. The speaker verification part is then based on a pool of Multi-Layer Perceptrons (MLPs) trained to discriminate between the client speaker and the world speakers.

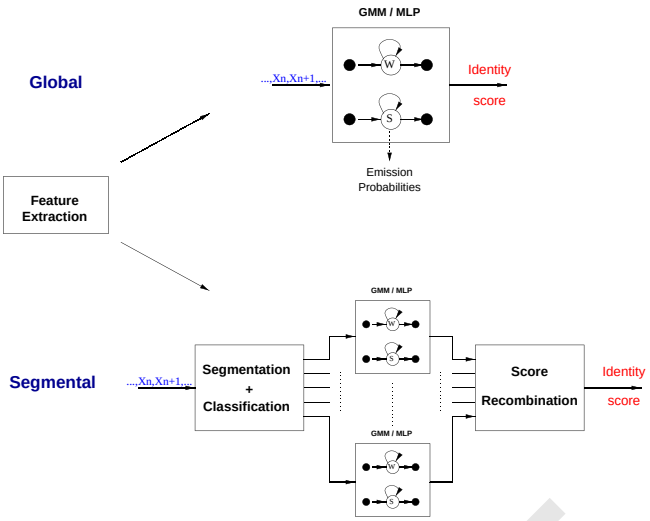

Fig. 6. Global and segmental speaker verification systems

We compare the performances of the ALISP based segmental speaker verification versus a similar global system on the NIST'98 corpus including 250 male and 250 female speakers. Classical text-independent Gaussian Mixture Model (GMM) based systems are used as the baseline system.

5.3 System Description.

Global Speaker Modeling. The classical way to do pattern classification in text-independent systems is to assign a unique probability density function (pdf) to the whole vector sequence. One way to build the pdf is to use Gaussian Mixture Models [34] in which the multivariate distribution is modeled with a weighted sum of Gaussian distributions.

Another way to perform classification is to use Artificial Neural Networks (ANNs) [20]. In previous studies, ANNs have successfully been used for speaker verification and we refer to [2] for a review of such approaches dedicated to speaker recognition. Amongst the different ANN architectures, Multi-Layer Perceptrons are often used. Their main potential advantages against GMM include, among others, discriminant capabilities, weaker hypotheses on the acoustic vector distributions and possibility to include a larger acoustic frame window as input to the classifier. The main drawback is that their optimal architecture has to be selected by trial-and-error procedures. For speaker verification purposes, Multi-Layer Perceptrons, one per client speaker, are discriminatively trained to distinguish between the client speaker and the background world speakers. Two outputs are generally used, one for the client and the other for the world class. If each output unit k of the MLP is associated to class category C_k, it is possible to train the MLP to generate a posteriori probabilities $p(C_k|x_n)$ [6]. During train-

ing, the parameters are iteratively updated via a gradient descent procedure in order to minimize the difference between the actual and desired outputs. Training is said to be discriminant because it minimizes the likelihood of incorrect models and maximizes the likelihood of the correct model.

In the case of global speaker modelling with GMM or MLP, the sequence of feature vectors is fed into a unique classifier that outputs a score for the client model and the world model, i.e. respectively S_c and S_w (see Fig. 6, top part), and the decision (reject/accept the speaker) is performed by comparing the ratio of the client and world scores to a threshold according to :

$$\log(S_c) - \log(S_w) > T \ \rightarrow \text{accept} \,, \tag{9}$$

$$\log(S_c) - \log(S_w) \leq T \ \rightarrow \text{reject} \,. \tag{10}$$

Segmental Speaker Modeling. In the segmental ALISP text-independent speaker modeling approach (see Fig. 6, lower part) the first step is to segment and label the speech into categories. Segmentation is achieved using temporal decomposition and the classification step is performed with vector quantization, as introduced in Sect. 3.3. In such a way each vector of the acoustic sequence is classified as a member of a category C_l determined through the segmentation and the labeling. In the modeling step, the same technique as for global modeling is used. L MLPs are trained for each client, where L is the number of codebook centroids. At test time, the test speech is also segmented into L categories and each category is tested against the corresponding MLP. In such a way the MLP associated with category C_l provides a segmental score as follows :

$$S_{cl} = \prod_{x \in C_l} P(M_{cl}|x)/P(M_{cl}) \,, \tag{11}$$

$$S_{wl} = \prod_{x \in C_l} P(M_{wl}|x)/P(M_{wl}) \,, \tag{12}$$

where the products involve vectors being previously labeled as members of category C_l. Subscripts cl and wl denote respectively the client model for category C_l and world model for the segmental C_l.

5.4 Speaker Verification Experiments

Task Description. Segmental and global systems are tested on the NIST'98 database, part of the SWITCHBOARD II Phase II corpus, recorded over telephone lines. The speech is spontaneous and no transcriptions, orthographic or phonetic, are available. The database consists of 250 male and 250 female speakers representing the clients and the impostors of the system. The gender mismatch is not studied, so that all experiences are strictly gender-dependent. Gender-dependent results are merged in a unique curve, for sake of simplicity. Only one training and testing configuration is considered: 2 min or more for the training and 30 s of speech for the test duration. To evaluate the robustness of the new proposed segmental method, some of the tests are evaluated

separately for matched and mismatched conditions (of the training and testing material). They are noted respectively as SN (same number) and DT (different microphone type). An independent set of 100 female and 100 male speakers with mixed carbon and electret microphones was selected from the NIST'97 database for modeling the world speakers. The experimental results are described as follows. First the global MLP performances are compared with the state of the art GMM based system. The influence of the mismatched training and testing conditions is pointed out, and the influence of the length of the acoustic window is discussed. Segmental results are described afterwards and some per-class performance details are given.

Experimental Setup. LPC-cepstral parameters are used for the feature extraction. A 30 ms Hamming window is applied every 10 ms in order to extract 12 LPC-cepstral coefficients. The order of the LPC analysis is set to 10. A liftering procedure is applied to the cepstral vectors followed by cepstral mean subtraction in order to reduce the effects of the channel. The MLP parameters, although not fully optimized, were experimentally tuned to reach acceptable performances for the different systems. MLPs used for the global systems have three layers with 120 neurons in the hidden layer and two output units. For the segmental MLPs, the number of neurons in the hidden layer is reduced to 20. In both cases, the input size of the MLP is defined by the number of contiguous frames set as input of the MLP. We use the notation Cxy to denote inputs with x frames to the left and y frames to the right of the central frame. The temporal decomposition is set to detect 15 events per second in average and the vector quantization is trained on the 1997 data with codebook size of $L = 8$. Coherence of the acoustic labeling among speakers is verified through informal listening tests. The speaker scores are normalized with the background world model and Z-normalisation is applied for each system [35].

ROC and DET Curves. Performances of speaker verification systems are usually given in terms of False Alarms and Miss Probability, often represented as Receiver Operating Curves (ROC). When similar systems need to be compared, it is more practical to use a Detection Error Tradeoff (DET) [26] representations, in which the x and y scales are normal deviate scales.

Global System Results. We first carried on the NIST'98 database several sets of experiments to investigate the effects of increasing the temporal information at the input Cxy of the MLP. Our previous studies [32] and [19] showed that the length of the temporal information used at the input of the MLP is of crucial importance for reaching acceptable classification performances. Fig. 7 demonstrates this behavior. The number of input frames spans from one (noted as $C00$, corresponding to 30 ms) to 11 frames (noted as $C55$, equivalent to 130 ms). Using more contiguous input frames improves the performances of the global MLP systems, however a saturation appears when eleven frames are

used as input. This result demonstrates the importance of the frame to frame temporal information for speaker verification experiments. Further experiments, not reported here, were conducted to determine the optimal number of hidden nodes in the hidden layer. In such a way, a $C55$ MLP with 120 hidden neurons was selected as our baseline global MLP reference.

Comparison performances of global MLP and GMM systems are shown in Fig. 8. Actually better results for speaker verification are achieved with GMMs that are used here as the state-of-the-art comparison point. This difference might come from the fact that one part of the training data (about 10%) is kept apart to avoid over-training of the MLP with a usual cross-validation procedure. Although GMMs perform slightly better than MLPs in the global case, the discriminant capability of MLPs make them better candidates for the segmental system and we adopted them for the segmental experiments.

The importance of the mismatched training and testing conditions, as far as the microphone differences are considered, are also visible on this figure. When the speech signal comes from a different handset type than the training speech material (DT curves), the error rates are increased roughly by a factor of five. These results point out the fact that microphone differences are one of the most serious obstacles to be solved for improving speaker recognition performances.

Segmental System Results. For the segmental system, the speech signal is first segmented and labelled into categories. One important factor is the amount of training material available per class. It is well known that the more training material we have, the better the models are. If automatically determined speech units correspond to phonemes, the number of classes should approximately be equal to the number of phonemes. However two minutes of training material is here not sufficient to ensure a proper training of all the classes. This is the reason why the number of classes is set to eight, so that broad phonetic classes are detected.

Discrimination between client and world speakers is then performed separetely on each category. Performances on a per-class basis for the segmental system are depicted in Fig. 9 for the same number (SN) condition. Only five out of eight classes are illustrated for sake of clarity. The DET curves clearly show that classes perform differently and convey more or less information about the speakers. A similar conclusion is reported in our previous studies [32], in which categories were directly obtained through forced alignment in phoneme models.

A score recombination of the independent classifiers is necessary to obtain a global decision for speaker verification. The differences of class performances, as reported in Fig. 9, suggest that the recombination of scores obtained for each category should be non-linear, giving more weight to the most discriminant classes. This issue is left apart in our analysis and a simple linear recombination of scores was used to obtain global decisions. Performances of this approach are reported in Fig. 10 where we compare the best global MLP system (noted as MLPGlob C55) and the segmental MLP system (noted as MLP SegC22 RLin). Although a simple linear recombination of individual scores has been used, the

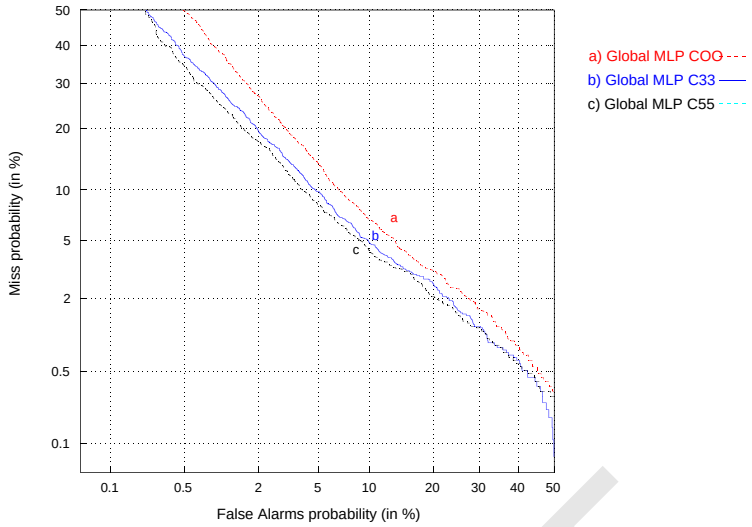

Fig. 7. DET curves for global MLP systems, showing the influence of the MLP input window length Cxy (varying from C00=30ms to C55=130ms). The segment duration are 2 min or more for training and 30 sec for testing

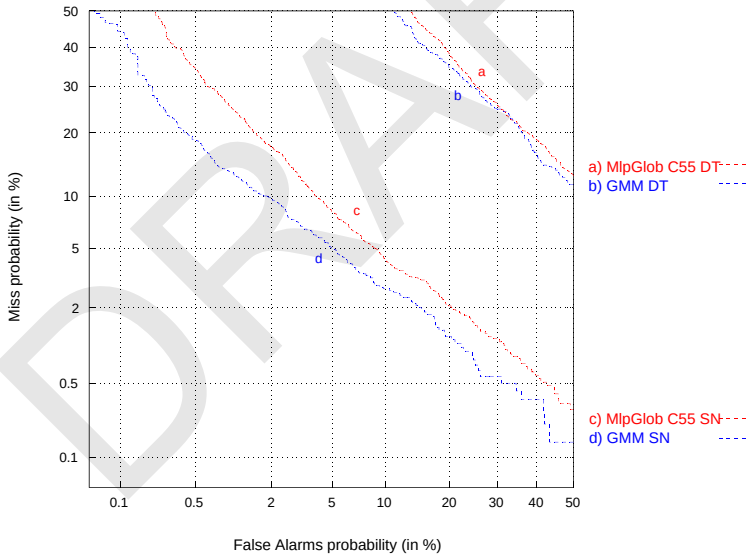

Fig. 8. DET curves for global GMM and MLP systems, showing the performances for test segments collected from same type microphones (SN) vs. different type microphones (DT). The segment duration are 2 min or more for training and 30 sec for testing

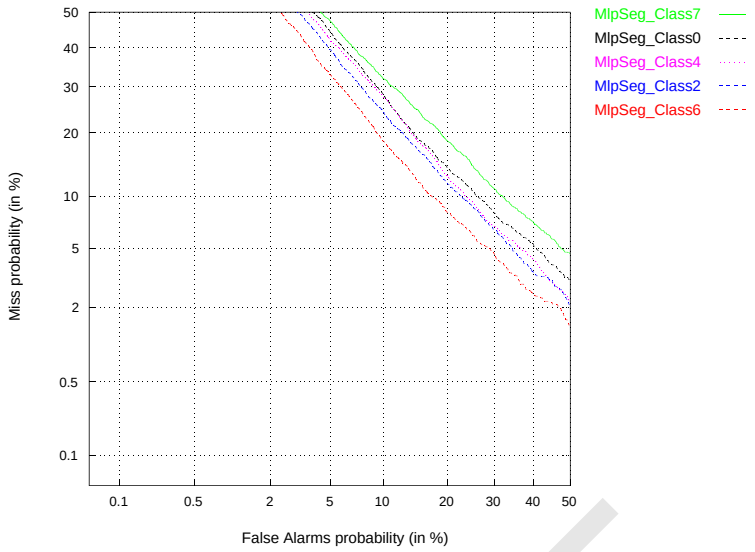

Fig. 9. DET curves for segmental MLP systems, showing the performances of five out of eight classes. The segment duration are 2 min or more for training and 30 sec for testing

segmental approach reaches almost equivalent performances as the global systems in the case of matched conditions and performs better than both MLP and GMM global systems in the case of mismatched conditions.

Many issues are still open with the segmental approach as proposed here. For example, per-class individual tuning of the parameters should be investigated (number of input frames, thresholding, normalization, etc.) and better score recombination taking into account the discriminant performances of categories should be analyzed. Nevertheless, even using simple strategies for producing the alignment (temporal decomposition) and for recombining individual scores, very good results are already reported on a standard evaluation task.

Our results show that ALISP techniques are potentially useful also in speaker verification. With our new segmental approach, based on ALISP segmentation of the speech signal and coupled with a MLPs for speaker classification, we reached comparable performances as state-of-the-art global system. For the "difficult" mismatched conditions (different type of microphone for training and testing), the segmental MLP system slightly ouperforms the global MLP and GMM system. These results are encouraging for pursuing the futher developments of such segmetnal systems.

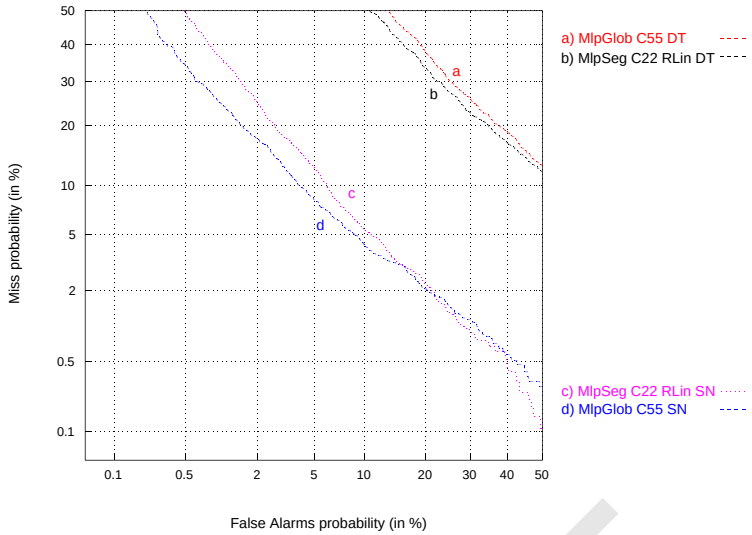

Fig. 10. DET curves for global and segmental MLP systems. MLPGlobC55 stands for the global system with 11 input frames and MLPSegC22RLin indicates the segmental system with linear score recombination using 5 input frames. Performances are reported for test segments collected from same type microphones (SN) vs. different type microphones (DT). The segment duration are 2 min or more for training and 30 sec for testing

6 Conclusion and perspectives

The rapid development of interactive voice servers in a multi-lingual environment calls for an intensive use of data-driven techniques to specify the acoustic units and models to be used by the recognizer and to train a dialogue manager. This chapter has proposed a set of tools which could be used for these purposes. The acoustic units have been evaluated in the context of a very low bit-rate coder. Such a coder could be either language independent or specific to a given language and a given speaker. Language dependent coders would help for language identification.

Many of the voice servers may perform better with some knowledge about the identity of the speaker. The recognizer could adapt to that speaker in the first place. Furthermore, for security purposes, it may be necessary to verify the identity of the user. This chapter proposes a promising ALISP-based approach to the speaker verification problem.

Acknowledgments

Some aspects of the work described in this chapter greatly beneficiated from the help of the following researchers and engineers: Frederic Bimbot provided us with

his software for temporal decomposition, Sabine Deligne developed the 'multigram' model, Frank Neubert contributed to the recognition of proper names and spellings and Achim Latz implemented the first version of the 'Majordome'.

This work was made possible with the financial help of the EEC, the Swiss OFES, the French government and Swisscom. The work of Jan Černocký was partly supported by the Ministry of Education, Youth and Sports of the Czech Republic, project No. VS97060.

References

[1] Atal, B.: Efficient coding of LPC parameters by temporal decomposition. Proc. IEEE ICASSP 83, (1983) 81–84

[2] Bennani, Y., Gallinari, P.: Connectionist approaches for automatic speaker recognition. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification, Martigny, Switzerland, (1994) 95–102

[3] Bimbot, F.: An evaluation of temporal decomposition. Technical report, Acoustic research departement AT&T Bell Labs, (1990)

[4] Bimbot, F., Deligne, S., Yvon, F.: Unsupervised decomposition of phoneme strings into variable-length sequences, by multigrams. ICPHS, Stockholm, (1995)

[5] Bimbot, F., Pierraccini, R., Levin, E., Atal, B.: Modèles de séquence à horizon variable : multigrammes. Actes des XXèmes journées d'études sur la parole, Trégastel, (1994) 467–472

[6] Bourlard, H., Wellekens, C.: Links between markov models and multi-layer perceptrons. IEEE Trans. Patt. Anal. Machine Intell. 12(12) (1990) 1167–1178

[7] Chollet, G., Cochard, J.L., Constantinescu, A., Jaboulet, C., Langlais, P.: Swiss French PolyPhone and PolyVar: Telephone speech databases to model inter- and intra-speaker variability. John NERBONNE, editor, Linguistic databases CSLI Publications (1997)

[8] Chollet, G., Černocký, J., Constantinescu, A., Deligne, S., Bimbot, F.: Towards ALISP: a proposal for Automatic Language Independent Speech Processing. In Keith Ponting, editor, NATO ASI: Computational models of speech pattern processing Springer Verlag, in press

[9] Cole, R., Roginski, H., Fanty, M.: English alphabet recognition with telephone speech. Eurospeech Proceedings (1991) 479–482

[10] Dedina, M.J., Nusbaum, H.C.: PRONOUNCE: a program for pronunciation by analogy. Computer Speech and Langage 5 (1991) 55–64

[11] Deligne, S.: Modèles de séquences de longueurs variables: Application au traitement du langage écrit et de la parole. PhD thesis École nationale supérieure des télécommunications (ENST) Paris (1996)

[12] Deligne, S., Sakisaga, Y.: Learning a syntagmatic and paradigmatic structure from language data with a bi-multigram model. Proceeding of COLING/ACL'98 Montral (1998) 300–306

[13] Deligne, S., Yvon, F., Bimbot, F.: Introducing statistical dependencies and structural constraints in variable-length sequence models. In Laurent Miclet and Colin de la Higuera, editors, Grammatical Inference: Learning Syntax from Sentences Lecture Notes in Artificial Intelligence 1147 Springer (1996) 156–167

[14] Dietterich, T.G., Hild, H., Bakiri, G.: A comparison of ID3 and backpropagation for English text-to-speech mapping. Machine Learning 18(1) (1995) 51–80

[15] Eatock, J.P., Mason, J.S.: A quantitative assessment of the relative speaker discriminant properties of phonemes. ICASSP 1 (1994) 133–136
[16] Fukada, T., Bacchiani, M., Paliwal Sagisaka, K.: Speech recognition based on acoustically derived segment units. Proc. ICSLP 96 (1996) 1077–1080
[17] Gorin, A.L., Riccardi, G., Wright, J.H.: How May I Help You? In Keith Ponting, editor, NATO ASI: Computational models of speech pattern processing. Springer Verlag, in press.
[18] Gravier, G. ,Etorre, G., Yvon, F., Chollet, G.: Directory name retrieval using HMM modeling and robust lexical access. Workshop on Automatic Speech Recognition and Understanding (1997)
[19] Hennebert, J., Petrovska-Delacréraz, D.: Phoneme based text-prompted speaker verification with Multi-Layer Perceptrons. RLA2C 98 Avignon France (1998) 55–58
[20] Hertz, J., Krogh, A., Palmer, R.G.: Introduction to the theory of Neural Computation Santa Fe Institute Studies in the Sciences of Complexity Addison Wesley (1991)
[21] Jouvet, D. et al.: Speaker-independent spelling recognition over the telephone. Int. Conf. on ASSP 2 (1993) 235–238
[22] Junqua, J.-C. et al.: An N-best strategy, dynamic grammars and selectively trained neural networks for real-time recognition of continuously spelled names over the telephone. Int. Conf. on ASSP (1995) 852–855
[23] Lennig, M.: Deploying large-scale speech recognition applications: experience from the field. 4th IEEE Workshop on Interactive Voice Technology for Telecommunication Applications (IVTTA) Torino September (1998)
[24] Loizou, P.C., Spanias, A.S.: High-performance alphabet recognition. IEEE Trans. on Speech and Audio Processing 4(6) November (1996) 430–445
[25] Luk, R., Damper, R.I.: Stochastic phonographic transduction for English. Computer Speech and Language 10 (1996) 133–153
[26] Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M.: The DET curve in assesment of detection task performance. Eurospeech Proceedings Rhodes Greece (1997) 1895–1898
[27] Meyer, M., Hild, H.: Recognition of spoken and spelled proper names. Eurospeech Proceedings (1997) 1579–1582
[28] Neubert, F., Gravier, G., Yvon, F., Chollet, G.: Directory name retrieval over the telephone in the PICASSO project. IVTTA (1998)
[29] Oflazer, K.: Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. Computational Linguistics, 22(1) (1996) 73–89
[30] Olsen, J.: A two-stage procedure for phone based speaker verification. In Borgefors, G., Bigün, J., Chollet, G., editor, First International Conference on Audio and Video Based Biometric Person Authentication (AVBPA) Crans Switzerland Springer Verlag Lecture Notes in computer Science 1206 (1997) 219–226
[31] Ostendorf, M., Price, P.J., Shattuck-Hufnagel, S.: The Boston University radio news corpus. Technical report Boston University (1995)
[32] Petrovska-Delacrétaz, D., Hennebert, J.: Text-prompted speaker verification experiments with phoneme specific MLPs. ICASSP Seattle (1998) 777–780
[33] Pye, D.: Automatic recognition of continuous spelled Swiss-German letters. Technical report IDIAP (1994)
[34] Reynolds, D.A.: Automatic speaker recognition using gaussian mixture speaker models. The Lincoln Laboratory Journal 8(2) (1995) 173–191

[35] Reynolds, D.A.: Comparison of background normalisation methods for text-independent speaker verification. Eurospeech Proceedings (1997) 963–966
[36] Schmid, P. et al.: Real-time, neural network-based, French alphabet recognition with telephone speech. Eurospeech Proceedings (1993) 1723–1726
[37] Schmidt, M., Fitt, S., Scott, T., ack, M.: Phonetic transcription standards for European names (ONOMASTICA). Eurospeech Proceedings 1 Berlin (1993) 279–282
[38] Sejnowski, T., Rosenberg, C.: Parallel networks that learn to pronounce English text. Complex Systems 1 (1987) 145–168
[39] van den Bosch, A.: Learning to pronounce written words: A study in inductive language learning. PhD thesis University of Maastricht (1997)
[40] Černocký, ., Baudoin, G., Chollet, G.: Segmental vocoder - going beyond the phonetic approach. Proc. IEEE ICASSP Seattle WA May (1998) 605–608
[41] Vitale, T.: An algorithm for high accuracy name pronunciation by parametric speech synthesizer. Computational Linguistics, 17(3) (1991) 257–276
[42] Yvon, F.: Grapheme-to-phoneme conversion using multiple unbounded overlapping chunks. Proceedings of the conference on New Methods in Natural Language Processing (NeMLaP II) Ankara Turkey (1996) 218–228
[43] Yvon, F.: Prononcer par analogie: motivation, formalisation et valuation. PhD thesis, Ecole Nationale Supérieure des Télécommunications (1996)