Towards Fully Automatic Speech Processing Techniques for Interactive Voice Servers

Gérard Chollet[1], Jan Černocký[2], Guillaume Gravier[1], Jean Hennebert*[3], 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
dijana.petrovska@epfl.ch

Abstract. Automatic Speech Processing (Speech Recognition, Coding, Synthesis, Language Identification, Speaker Verification, Interpreting Telephony,...) 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') focusses 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 create 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 telephone *Interactive Voice Servers* (IVS). Such servers interact with the caller using speech

* Currently at Ubilab, UBS IT Innovation Laboratory, Switzerland

input (recognition and understanding) and output (synthesis). For some applications (telephone card, banking, ...), 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, the syntax they can interpret, the future calls for 'Unrestricted Vocabulary Continuous Speech Recognition' for any speaker, any language, any dialect, 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[1] speaker verification evaluation campaigns.

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

[1] NIST organizes every year an evaluation of speaker verification systems. A unique data set and evaluation protocol are provided to each participating laboratory, so that intra- and inter-laboratory algorithms comparisons are significantly easier.

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 preferred 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, offers many of the possible 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,...), but takes a long time to reach the recipient. On the other hand, electronic, voice and fax mail are delivered almost instantaneously. Voice mail, 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 (although Optical Character Recognition (OCR) or retyping is necessary in order to use the code and not just to read it), 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 Personal Digital Assistant capable of connecting to one's mailbox via 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 OCR 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[2], Portico[3]) are being commercialized. They use proprietary hardware and software to be shared by multiple users. A more individualized 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 communication with registered users.

Another advantage of ISDN boards is the fact that multiple applications can gain 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. The ISDN board is programmed using Microsoft Visual C++ 5.0 Professional and the CAPI, the Common ISDN Application Programming Interface, a now widely accepted and OS 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

[2] http://www/mrtramp.com/Wildfire.htm

[3] http://www.generalmagic.com

accessible using this kind of API. Anyway, this approach is far better than writing a program for especially one 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 OS 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 that, 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 all languages and dialects of the world. 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. Section 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. The National Security Agency (NSA) in the United States has mandated the National Institute of Standards and Technology (NIST) to organize annual evaluation campaigns concerned with speaker verification. The last section of this chapter reports on our participation to these evaluations using some of the ALISP tools.

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 kinds of discovery procedure, in order to more accurately model 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*, ...; 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 application, 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],.... 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, ... 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.

First, the goal of *temporal decomposition* (TD) 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* (IF)

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

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 IF matrix. IF and targets are iteratively locally refined by minimizing the distance between $\mathbf{X}$ and $\hat{\mathbf{X}}$. Intersections of interpolation functions defines speech segments.

Then, *unsupervised clustering* assigns segments to classes. Vector quantization (VQ) is used for automatic determination of classes. The VQ codebook $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_L\}$ is trained by K-means algorithm with binary splitting. 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 (MG) [11] may serve for finding characteristic sequences of quantized TD events or of segments determined by HMMs. The method is based on finding optimal segmentation of 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 MGs. The likelihood is given by the product of probabilities $\mathcal{P}(x_i)$ of MGs 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, *Hidden Markov models* (HMM) can be used to model the units. HMM parameters are initialized using context-free and context-dependent Baum-Welch training with TD+VQ or TD+VQ+MG transcriptions, and refined in successive steps of corpus segmentation using HMMs and model parameters re-estimation. The speech represented by the observation vector string $\mathbf{O}$ can then be aligned with models by maximizing the likelihood

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

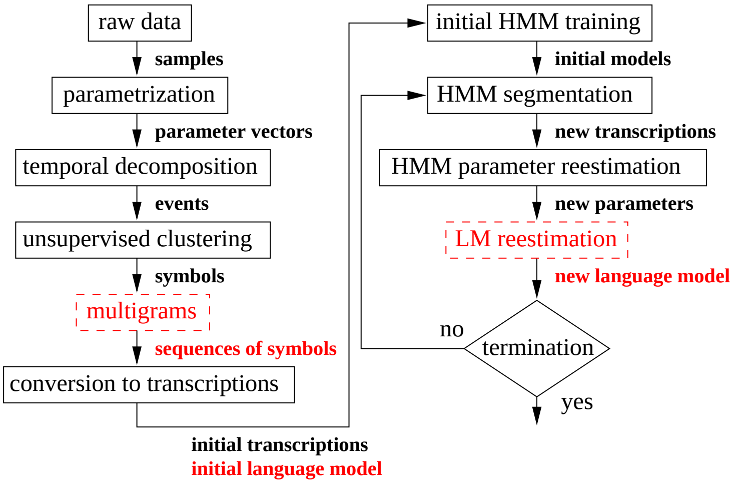

Fig. 1. Data-driven derivation of coding unit set in VLBR phonetic vocoder

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 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 acoustic-phonetic structure of the message and that it is appropriate also in other domains. Moreover (in contrast with 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 (CU) 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) supposing 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 are summarized in Tab. 1. In the first case, the synthesis was done by a simple concatenation of representative signals. In Boston University (BU) experiments, the synthesis was LPC-based 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.

Table 1. Summary of VLBR coding experiments

database language speakers	PolyVar Swiss French 1 (the most represented)	BU Radio Speech Corpus American English 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	1514	722 (F2B), 972 (M2B)
representatives per CU	8	8
R_u [bps] (test set)	120	110 (F2B), 119 (M2B)

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 obtained at BU using a segmental HMM recognizer constrained by possible pronunciations of 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 ALISP unit with a phoneme. The results are summarized in a confusion matrix $\mathbf{X}$ $(n_p \times n_a)$, whose elements are defined

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

where n_p and n_a are respectively the sizes of phoneme and 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 ALISP unit a_j. The columns of $\mathbf{X}$ are rearranged to let the matrix have a quasi-diagonal form[4] and the resulting matrix is given in Fig. 2. On contrary 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] bring interesting insights on this representation.

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

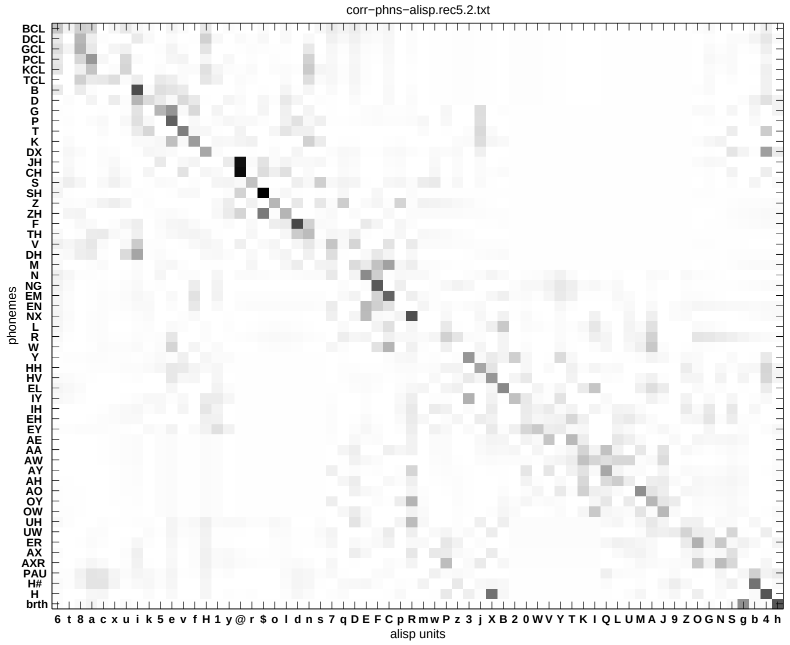

Fig. 2. Correspondence of ALISP segmentation and phonetic alignment for speaker F2B in BU corpus. White color corresponds to zero correlation, black to maximum value $x_{i,j}$=0.806

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 section 2) critically rely on its ability to handle proper names properly. 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, ...

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], [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 lexical search. These studies however fail to address a number of problems that are critical in the perspective of real applications. First, they greatly underestimate the variability observed in real-life spellings. Second, 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]. Section 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, as a conclusion we explain in Sect. 4.4 how multigrams can be extended to induce the pronounciation 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*"...

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

Acoustic modeling. To recognize the pronunciations, phone models are necessary 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", ...) 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 first automatically segmented from its orthographic transcription, using task-independent phoneme models, trained on sentences of the Swiss-French Polyphone database [7][5], 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 improve 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.

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

[5] Distributed by ELRA http://www.icp.inpg.fr/ELRA

training corpus. The same procedure is used to determine insertion and deletion costs. This approach is somewhat similar to HMM based alignment procedure 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 belongs 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 conforms 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 acoustic modeling. Finally, 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. Figure 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 LM 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

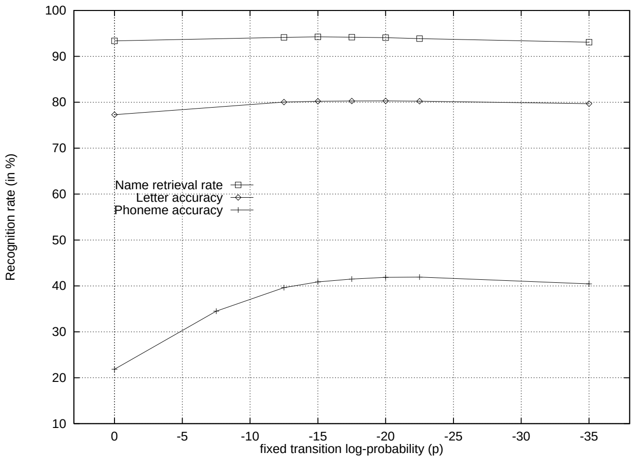

Fig. 3. Decoding accuracy on the train corpus as a function of the fixed transition log-probability p

Table 2. Recognition rate (in %)/accuracy 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

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 LM significantly improves the quality of the phone decoding but letter recognition still remains more reliable.

Finally, 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],[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 substitution 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

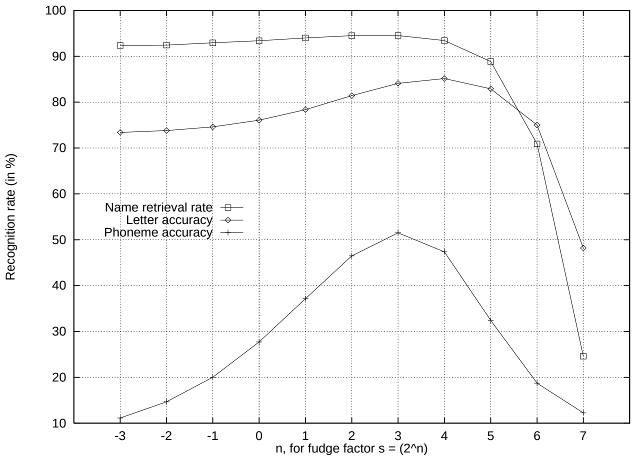

Fig. 4. Decoding accuracy on the train corpus as a function of the fudge factor

with recognition rates of respectively 84.0% and 79.1% at the optimal point. When using a LM 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.

The search strategy presented before implies an exhaustive search across the entire lexicon and 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 pre-selection of 100 lexical entries with a very small decrease of the performances (83.2% instead of 83.6%).

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

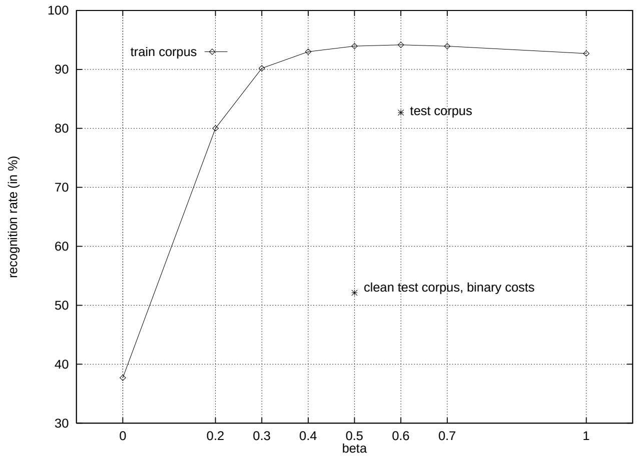

Fig. 5. Name recognition rate as a function of β

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 kinds of assumptions, seem to fit in better with our general principles: language independence, use of acoustic or linguistic units for representing pronunciations. In line with the core ideas of ALISP, we develop hereafter yet 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 bidimensional streams.

The basic idea of the joint-multigram model is to automatically identify recurrent joint sequences in a pronunciation dictionary, and use these 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 respectively 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 border 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 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 program is rewritten:

$$
\begin{aligned}
\widehat{\Omega}^* &= \arg\max_{\Omega} \;\; \mathcal{L}(O,\ \Omega,\ L_O^*,\ L_\Omega^*) && (6)\\
&= \arg\max_{\Omega} \;\; \mathcal{L}(O,\ L_O^* \mid \Omega,\ L_\Omega^*)\, \mathcal{L}(\Omega,\ L_\Omega^*) && (7)
\end{aligned}
$$

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

$$
\widetilde{\Omega}^* = Argmax_{\Omega} \;\; \mathcal{L}(O,\ L_O^* \mid \Omega,\ L_\Omega^*)\;\; \widetilde{\mathcal{L}}(\Omega,\ L_\Omega^*)\;\; . \qquad (8)
$$

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 consists 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 fair recognition performances with small amount of speech signal needed for training and testing. The major drawbacks of such systems are the poor security level (the system can easily be fooled using pre-recorded speech) and their repetitious nature.

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 passwords is used. As well, 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 systems 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 can be more precise. 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 fusion 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. ALISP offers an alternative when no annotated training data are available and could potentially boost up the performances. These are the reasons that led us to investigate a text-independent segmental approach based on ALISP techniques. Temporal decomposition followed by vector quantization is used to obtain classes of sounds. The speaker verification part is based on multi-layer perceptrons (MLP) trained to discriminate between the client speaker and world speakers [2].

We compare the performances of the segmental speaker verification versus a similar global system on the NIST 1998 corpus including 250 male and 250 female speakers. Classical text-independent Gaussian Mixture Model (GMM) based systems are used as the baseline system [34].

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 in which the multivariate distribution is modeled with a weighted sum of Gaussian distributions.

Another way to perform classification is to use Artificial Neural Nets [20]. Multi-layer perceptrons are often used. They include discriminant capabilities and weaker hypotheses on the acoustic vector distributions. The main drawback is that their optimal architecture must be selected by a trial and error procedure. The Multi-layer Perceptrons, one per client speaker, are discriminatively trained to distinguish between the client speaker and a background world model. 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 categories C_k, it is possible to train the Artificial Neural Net to generate a posteriori probabilities $p(C_k|x_n)$ [6]. During training, the parameters are iteratively updated via a gradient descent procedure in order to minimize the difference between the actual and desired

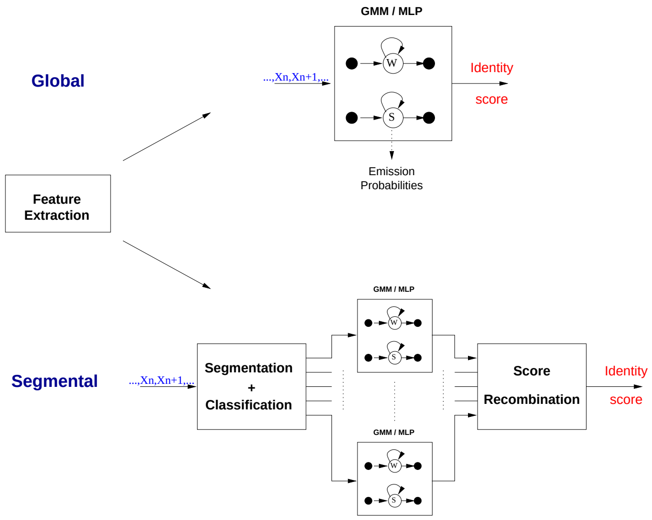

Fig. 6. Global and segmental speaker verification systems

outputs. Training is said to be discriminant because it minimizes the likelihood of incorrect models and maximizes the likelihood of the correct model.

For GMM as well as for 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 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 text-independent speaker modeling approach (see Fig. 6, bottom 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 :

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

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

where the products involve vectors being previously labeled as members of category C_l. Subscripts cl and wl denote respectively the client model for segmental category C_l and world model for segmental category 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, neither orthographic nor phonetic, are available. The database consists of 250 male and 250 female speakers representing the clients and the impostors of the system. The sex mismatch is not studied, so that all experiences are strictly sex-dependent. Sex-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. In the next section the influence of the length of the acoustic window is discussed. These experiments provide the necessary comparison points for the segmental system results described afterwards, where the performances per class are detailed. Finally, results with a simple recombination technique 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 structure of the MLPs used for the global systems is a three layer MLP, and with 120 neurons in the hidden layer. For the segmental MLPs, the number of neurons in the hidden layer is reduced to 20 and 5 contiguous frames are used as the input for the MLPs. 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. Z-norm 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

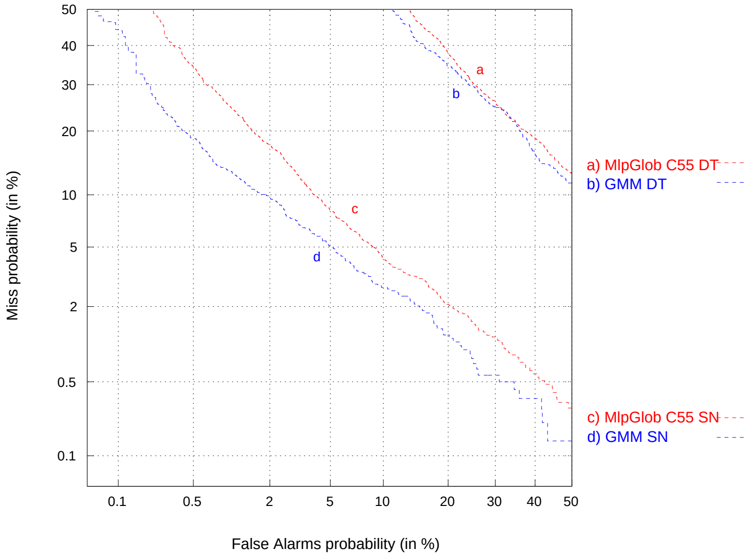

Fig. 7. Global systems, GMM and MLP modeling, training condition 2 min or more, test duration 30 sec, same number (SN) and different type (DT), for train and test materials

Receiver Operating Curves (ROC). When similar systems need to be compared, it is more practical to use a Detection Error Tradeoff (DET) [26] in which the x and y scales are normal deviate scales.

Global System Results. Actually better results for speaker verification are achieved with GMMs. We use them here as the state of the art comparison point. The comparisons of the performances of the global MLP and GMM systems are shown in Fig. 7. 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 test segments come from a different handset type than the training speech material (DT curves), the error rates are increased roughly by a factor of four. Global GMM and MLP have comparable results. Taking into account the further discriminant possibilities we can have with MLP, we adopted them for the segmental experiments. It is known that one important factor for speaker verification is the amount of training data. For the MLP based experiments, one part of the training data (usually 10) is used for the cross-validation during the MLP training procedure. So the different performance of these two systems is perhaps due to the smaller train data of the MLPs.

Influence of the Input Window Length. Our previous studies [32] and [19] showed the importance of the acoustical window length used as the input of the MLP for speaker verification experiments. Fig. 8 demonstrates the behavior of

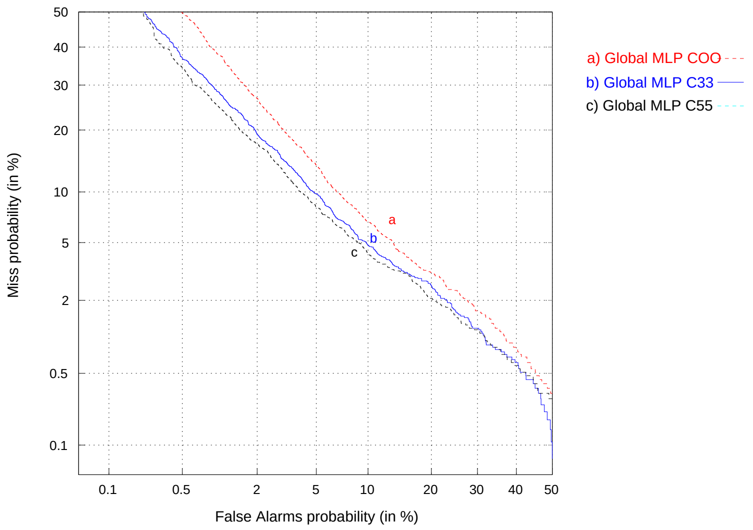

Fig. 8. Global system, MLP modeling, training condition 2F (2 min or more), test duration 30 sec, same number, influence of the input training window length (varying from C00=30ms to C55=130ms)

the MLP with different input window lengths. The number of input frames spans from one (noted as C00, corresponding to 30 ms) to 11 input 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.

Segmental System Results. Performances on a per-class basis for the segmental system (SN conditions) are depicted in Fig. 9. Only five classes having dissimilar performances are chosen for illustration. Classes perform differently and convey more or less informations about the speakers. 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. In the case when the automatically determined speech units are supposed to correspond to phonemes, the number of classes should approximately be equal to the number of phonemes. However two minutes of training material might not be 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. When using fusion techniques to recombine the scores of all these classes, one should use the information that certain classes perform better than others.

In Fig. 10 we compare the best global MLP system and the segmental MLP system. The segmental results are obtained through simple recombination of the eight classes (noted as MLP SegC22 RLin). With this simple recombination technique we observe a slight degradation for the same number conditions. For

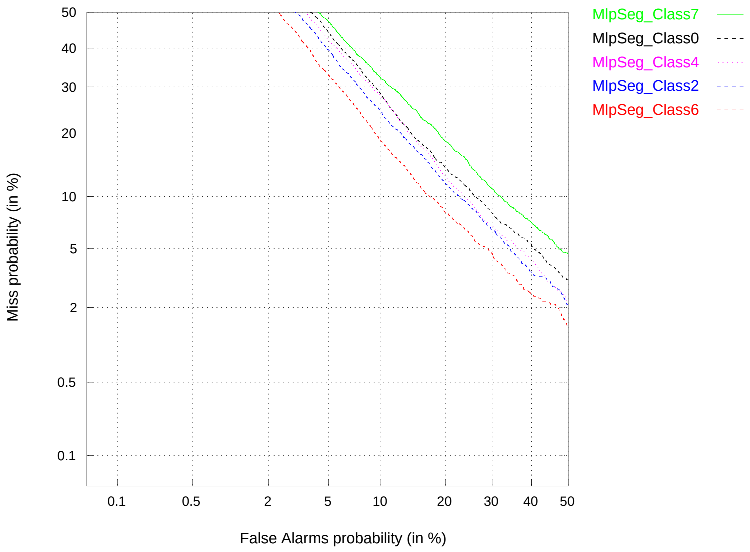

Fig. 9. Global (MlpGlob) and segmental (MlpSeg) systems, training condition 2 min or more, test duration 30 sec, same number (SN) and different type (DT) conditions. RLin indicates a linear recombination technique

the more difficult (different type) condition, the segmental system outperforms the global MLP system, and even the GMM results. This opens the way to fusion techniques, where individual tuning of parameters corresponding to each class can be done.

The proposed segmental system (automatic segmentation performed by temporal decomposition and vector quantization is, coupled with artificial neural network scoring) reaches similar performances as the global MLP system, and even outperforms it in mismatched training/test conditions. We show that ALISP techniques are potentially useful also in speaker verification because they are automatic and unsupervised, limiting the human interaction necessary, and hence the number of errors introduced by human operators. Two issues are still open regarding the segmental approach. First, per-class individual tuning of the parameters should be investigated (thresholding, normalization, . . .). Second, better merging of the class-dependent results to obtain the global scores, taking into account the discriminant performances of the classes should be analyzed.

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.

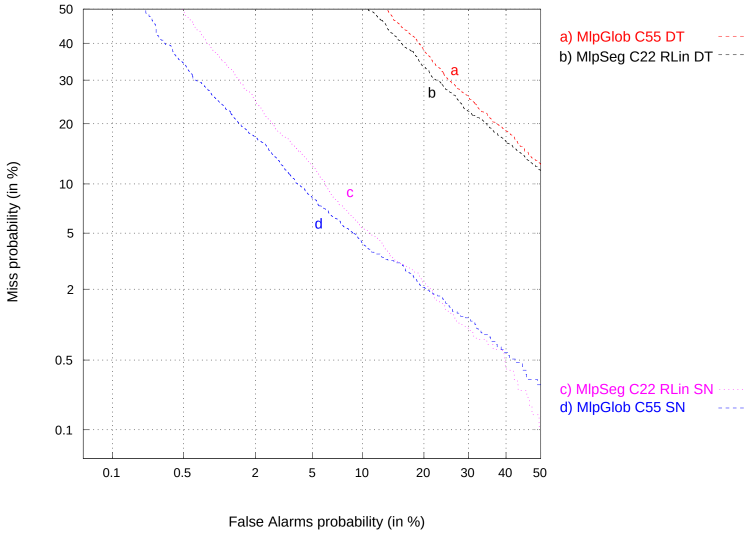

Fig. 10. Global and segmental system, training condition 2F, test duration 30 sec, same number (SN) and different type (DT)

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 reviews the state of the art in text dependent, text prompted and text independent speaker verification and proposes an ALISP-based approach to the 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] B. S. Atal. Efficient coding of LPC parameters by temporal decomposition. In *Proc. IEEE ICASSP 83*, pages 81–84, 1983.

[2] Younès Bennani and Patrick Gallinari. Connectionist approaches for automatic speaker recognition. In *ESCA Workshop on Automatic Speaker Recognition, Identificatio n and Verification*, pages 95–102, Martigny, Switzerland, April 1994.

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

[4] Frédéric Bimbot, Sabine Deligne, and François Yvon. Unsupervised decomposition of phoneme strings into variable-length sequences, by multigrams. In *ICPHS*, Stockholm, 1995.

[5] Frédéric Bimbot, Roberto Pierraccini, Esther Evin, and Bishnu Atal. Modèles de séquence à horizon variable : multigrammes. In *Actes des XXèmes journées d'études sur la parole*, pages 467–472, Trégastel, 1994.

[6] Hervé Bourlard and C. J. Wellekens. Links between markov models and multilayer perceptrons. *IEEE Trans. Patt. Anal. Machine Intell.*, 12(12):1167–1178, December 1990.

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

[8] G. Chollet, J. Černocký, A. Constantinescu, S. Deligne, and F. Bimbot. 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] Ronald Cole, Krist Roginski, and Mark Fanty. English alphabet recognition with telephone speech. In *EuroSpeech*, pages 479–482, 1991.

[10] Michael J. Dedina and Howard C. Nusbaum. PRONOUNCE: a program for pronunciation by analogy. *Computer Speech and Langage*, 5:55–64, 1991.

[11] S. Deligne. *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] Sabine Deligne and Yoshinori Sakisaga. Learning a syntagmatic and paradigmatic structure from language data with a bi-multigram model. In *Proceeding of COLING/ACL'98*, pages 300–306, Montréal, 1998.

[13] Sabine Deligne, François Yvon, and Frédéric Bimbot. 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, pages 156–167. Springer, September 1996.

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

[15] J. P. Eatock and J. S. Mason. A quantitative assessment of the relative speaker discriminant properties of phonemes. In *ICASSP*, volume 1, pages 133–136, 1994.

[16] T. Fukada, M. Bacchiani, and K. Paliwal Y. Sagisaka. Speech recognition based on acoustically derived segment units. In *Proc. ICSLP 96*, pages 1077–1080, 1996.

[17] A. L. Gorin, G. Riccardi, and J. H. Wright. How May I Help You? In Keith Ponting, editor, *NATO ASI: Computational models of speech pattern processing.* Springer Verlag, in press.

[18] G. Gravier, G. Etorre, F. Yvon, and G. Chollet. Directory name retrieval using HMM modeling and robust lexical access. In *Workshop on Automatic Speech Recognition and Understanding*, 1997.

[19] J. Hennebert and D. Petrovska. Phoneme based text-prompted speaker verification with multi-layer perceptrons. In *RLA2C 98*, pages 55–58, Avignon, France, 1998.

[20] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the theory of Neural Computation.* Santa Fe Institute Studies in the Sciences of Complexity. Addison Wesley, 1991.

[21] D. Jouvet et al. Speaker-independent spelling recognition over the telephone. In *Int. Conf. on ASSP*, volume 2, pages 235–238, 1993.

[22] Jean-Claude Junqua et al. An N-best strategy, dynamic grammars and selectively trained neural networks for real-time recognition of continuously spelled names over the telephone. In *Int. Conf. on ASSP*, pages 852–855, 1995.

[23] M. Lennig. Deploying large-scale speech recognition applications: experience from the field. In *4th IEEE Workshop on Interactive Voice Technology for Telecommunication Applications (IVTTA)*, Torino, September 1998.

[24] Philipos C. Loizou and Andreas S. Spanias. High-performance alphabet recognition. *IEEE Trans. on Speech and Audio Processing*, 4(6):430–445, November 1996.

[25] Robert W.P Luk and Robert I. Damper. Stochastic phonographic transduction for English. *Computer Speech and Language*, 10:133–153, 1996.

[26] Alvin Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assesment of detection task performance. In *Eurospeech 1997*, pages 1895–1898, Rhodes, Greece, 1997.

[27] Michael Meyer and Hermann Hild. Recognition of spoken and spelled proper names. In *EuroSpeech*, pages 1579–1582, 1997.

[28] F. Neubert, G. Gravier, F. Yvon, and G. Chollet. Directory name retrieval over the telephone in the PICASSO project. In *IVTTA*, 1998.

[29] Kemal Oflazer. Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. *Computational Linguistics*, 22(1):73–89, 1996.

[30] J. Olsen. A two-stage procedure for phone based speaker verification. In G. Borgefors J. Bigün, G. Chollet, editor, *First International Conference on Audio and Video Based Biometric Person Authentication (AVBPA)*, pages 219–226, Crans, Switzerland, 1997. Springer Verlag: Lecture Notes in computer Science 1206.

[31] M. Ostendorf, P.J. Price, and S. Shattuck-Hufnagel. The Boston University radio news corpus. Technical report, Boston University, February 1995.

[32] D. Petrovska and J. Hennebert. Text-prompted speaker verification experiments with phoneme specific MLPs. In *ICASSP*, pages 777–780, Seattle, 1998.

[33] David Pye. Automatic recognition of continuous spelled Swiss-German letters. Technical report, IDIAP, 1994.

[34] Douglas Reynolds. Automatic speaker recognition using gaussian mixture speaker models. *The Lincoln Laboratory Journal*, 8(2):173–191, 1995.

[35] Douglas A. Reynolds. Comparison of background normalisation methods for text-independent speaker verification. In *Eurospeech*, pages 963–966, 1997.

[36] P. Schmid et al. Real-time, neural network-based, French alphabet recognition with telephone speech. In *EuroSpeech*, pages 1723–1726, 1993.

[37] Mark Schmidt, Sue Fitt, Tina Scott, and Mervyn Jack. Phonetic transcription standards for European names (ONOMASTICA). In *Eurospeech*, volume 1, pages 279–282, Berlin, sep. 1993.

[38] Terrence J. Sejnowski and Charles R. Rosenberg. Parallel networks that learn to pronounce English text. *Complex Systems*, 1:145–168, 1987.

[39] Antal van den Bosch. *Learning to pronounce written words: A study in inductive language learning.* PhD thesis, University of Maastricht, 1997.

[40] J. Černocký, G. Baudoin, and G. Chollet. Segmental vocoder - going beyond the phonetic approach. In *Proc. IEEE ICASSP 98*, pages 605–608, Seattle, WA, May 1998.

[41] Tony Vitale. An algorithm for high accuracy name pronunciation by parametric speech synthesizer. *Computational Linguistics*, 17(3):257–276, sep. 1991.

[42] François Yvon. Grapheme-to-phoneme conversion using multiple unbounded overlapping chunks. In *Proceedings of the conference on New Methods in Natural Language Processing (NeMLaP II)*, pages 218–228, Ankara, Turkey, 1996.

[43] François Yvon. *Prononcer par analogie: motivation, formalisation et évaluation.* PhD thesis, Ecole Nationale Superieure des Télécommunications, 1996.