DESIGN AND EVALUATION OF ACOUSTIC AND LANGUAGE MODELS FOR LARGE SCALE TELEPHONE SERVICES

A. Facco [b] D. Falavigna [b,*] R. Gretter [b] M. Viganò [a]

[a] *Reitek Spa, Viale Monza 265, 20126 Milano, Italy*

[b] *ITC-Irst, Via Sommarive 18, 38050 Povo, Trento Italy*

Abstract

This paper describes the specification, design and development phases of two widely used telephone services based on automatic speech recognition. The effort spent for evaluating and tuning these services will be discussed in detail.

In developing the first service, mainly based on the recognition of "alphanumeric" sequences, a significant part of the work consisted in refining the acoustic models. To increase recognition accuracy we adopted algorithms and methods consolidated in the past over broadcast news transcription tasks. A significant result shows that the use of task specific context dependent phone models reduces the word error rate by about 40% relative to using context independent phone models. Note that the latter result was achieved over a small vocabulary task, significantly different from those generally used in broadcast news transcription.

We also investigated both unsupervised and supervised training procedures. More-over, we studied a novel partly supervised technique that allows us to select in some

"optimal" way the speech material to manually transcribe and use for acoustic model training. A significant result shows that the proposed procedure gives performance close to that obtained with a completely supervised training method.

In the second service, mainly based on phrase spotting, a wide effort was devoted to language model refinement. In particular, several types of rejection networks were studied to detect out of vocabulary words for the given task; a major result demonstrates that using rejection networks based on a class trigram language model reduces the word error rate from 36.7% to 11.1% with respect to using a phone loop network. For the latter service, the benefits and related costs brought by regular grammars, stochastic language models and mixed language models will be also reported and discussed.

Finally, notice that most of experiments described in this paper were carried out on field databases collected through the developed services.

Key words: Automatic telephone services, Acoustic model training, Supervised/unsupervised/partly supervised training, Language model refinement.

1 Introduction

Currently, most of the research and technology developed within the speech recognition community is evaluated and compared on voice databases properly designed for benchmark tests, such as the ones selected in the DARPA competitions (Young and Chase, 1998; Pallet et al., 1999; Graff, 2002; Woodland, 2002). Although this allows us to measure the effectiveness of new models and

* Tel: +39(0)461-314562; fax: +39(0)461-314591

Email addresses: `facco@itc.it` (A. Facco), `falavi@itc.it` (D. Falavigna), `gretter@itc.it` (R. Gretter), `m.vigano@reitek.com` (M. Viganò).

algorithms in the field of Automatic Speech Recognition (ASR), very little work has been published on determining the same effectiveness upon data collected by means of real telephone services; as far as we know, extensive tests have been published only on the "How May I Help You" service, developed by AT&T (Gorin et al., 1997). Actually, most of the applications required by the telephone market do not face tasks similar to the DARPA ones (e.g., broadcast news transcriptions, spontaneous dialog transcriptions, etc.); more often, tasks accomplished by automatic services are much simpler, such as recognizing, or spotting, a small number of words or sentences. Nevertheless, speech recognition "accuracy" and "robustness" with respect to system input variabilities, caused by e.g., user behaviors, background noise, transmission channel distortion, etc., are crucial requirements of field services.

This paper discusses the effort spent to design and refine two widely used telephone services, and gives the corresponding performance.

The first one of these two services (see section 2.1.1 for a detailed description) is mainly based on the recognition of "alpha-digits": the effectiveness of some technologies, mostly derived from broadcast news transcription tasks, will be demonstrated upon field data collected with the service itself. More specifically, both the benefits and costs provided by a "partly supervised" training method (Facco et al., 2004), as well as by the use of context dependent phone models (Young et al., 1994), will be reported and discussed.

In the past, "lightly supervised" training was investigated (Lamel et al., 2001a, 2000, 2002) using close, but not exact, transcriptions of speech data. Using the procedure described in Lamel et al. (2001a), the authors concluded that "recognition results obtained with acoustic models trained on large quantities

of automatically annotated data are comparable (under 10% relative increase in the word error rate) to results with acoustic models trained on large quantities of data with detailed manual transcriptions". The same authors (Lamel et al., 2001b) also proposed a technique to adapt a general purpose acoustic model to task specific data.

In other works, Kamm and Meyer (Kamm and Meyer, 2001, 2004) investigated a method based on a simplified version of the boosting technique (Freund and Schapire, 1997) which allows us to iteratively select from a speech database, that was entirely transcribed, the subset corresponding to the highest word error rate. The authors, after having retrained on this subset, measured a decrease in the relative word error rate of 8.7% using only 35% of the available data; in the same studies, they suggest applying the method for selecting the data to transcribe from a given untranscribed database.

The procedure proposed in this paper (see section 3.2) uses rejection networks to discard, from a given speech database, the "less useful" set of files. The part of data that is retained is then employed, similarly to (Kamm and Meyer, 2001, 2004), to train acoustic models in both an unsupervised and supervised way. A major result indicates that the employment of the procedure in a "partly supervised" fashion, i.e. transcribing manually a small, carefully selected part of a task specific database and training over it, allows the system to achieve performance comparable with that obtained using a "completely supervised" method, which requires a much larger manual transcription effort.

Although context dependent phone models proved to be effective for large vocabulary recognition tasks (Odell, 1995; Young et al., 1994; Woodland, 2002), there is no clear evidence about their effectiveness over small vocabulary tasks.

Therefore, we carried out experiments using data acquired with the developed telephone service, measuring a significant performance improvement when context dependent models are used to recognize an alpha-digit task.

The second investigated service is a voice portal mainly based on "phrase spotting". This service has to handle a simple dialog interaction with callers, asking for the desired information and for the related confirmations. Even if the task is simple, the design of such an application is rather critical and requires particular care both in defining the voice prompts and in designing the speech recognition grammars. For instance, during the development of a different service in the past years (Falavigna and Gretter, 1997), we observed that user answers to the voice prompt *"say yes or no"* were strictly *"yes"* or *"no"* only in 44% of the cases. This percentage rose to 67% when the voice prompt was changed to *"say* **clearly** *yes or no"*.

At present, there are no commonly accepted methods to design dialog applications. The design process often consists in defining a set of goals for a given application domain, and in implementing a "dialog strategy", mainly specified by a set of rules aimed at achieving the goals. Examples of rules that define a dialog strategy are activation of voice recognition grammars, activation of voice prompts, retrieval of information from databases, etc.

Although encouraging, the results based on probability theory (Puterman, 1994) to estimate the dialog strategy, obtained in recent works (Levin et al., 2000; Young, 2002; He and Young, 2003), are still premature to be used to develop real applications. Therefore, to design the above mentioned voice portal application, we decided to use the "frame based" approach reported in (Falavigna and Gretter, 1998). Actually, our experience suggests that, from a purely

practical point of view, the rapid and successful development of dialog applications still relies on the use of modular architectures, based on components easily reusable across different domains. For example, basic speech recognition grammars, such as "dates", "hours", "name entities", "numbers", "amounts", etc., can be exploited for accomplishing elementary subgoals in different application domains. Furthermore, dialog models similar to the one we adopted allowed researchers to develop several successful prototype systems in the past (Levin and Pieraccini, 1995; Ward and Issar, 1995; Zue and et al., 2000), as well as some real services (Harald and Schroer, 1998).

The main features of the dialog engine used in this work are summarized below.

The mixed language model proposed in (Falavigna et al., 2000) makes it possible to improve grammar effectiveness and portability, as well as to discard phrases not covered by speech recognition grammars e.g., by means of rejection networks or confidence measures (Gretter and Riccardi, 2001). As a matter of fact, the relative word error rate was reduced by more than 50% using a trigram language model to design the rejection networks employed in the voice portal service.

The capability of recognizing speech during a voice message playback (barge-in) adds flexibility to services.

In the same way, the ability to detect and possibly correct errors (Orlandi et al., 2003a) improves both robustness and flexibility.

This paper is organized as follows. The telephone architecture, together with the service specifications and design, is described in section 2. Acoustic and

language models, together with procedures and algorithms utilized for their training and/or refinement, are described in section 3. Experiments and results, obtained on field speech databases collected during the use of the services themselves, are reported in section 4. Finally, section 5 concludes the paper.

2 Telephone infrastructure and service design

The system manages speech interactions during communications by means of both a multi-client and a multi-server architecture. The telephone infrastructure is formed by the telephone front-end, consisting of some commercially available telephone boards, one or more speech servers and the enterprise back-end. The telephone front-end manages the user interface. Speech servers perform ASR and Text-To-Speech (TTS) functions. The enterprise back-end hosts database and legacy applications. Speech resources can be concurrently accessed by several client applications located on one or more platforms. Resource allocation is controlled by a centralized manager, implementing load balancing algorithms, and providing redundancy mechanisms for achieving high availability and scalability. Other main features are barge-in capability and VoiceXML support.

2.1 Service specifications

The two telephone services examined here are (1) a speech IVR application, delivered by Automobile Club Italia (**ACI**), used for the automatic payment of the road tax; and (2) a banking voice portal, delivered by **Unicredit** bank, used for accessing some existing customer services. The former has mainly to

deal with the recognition of utterances consisting of alphanumeric sequences: in order to reach a satisfactory level of performance on this task, a significant effort was devoted to refine the acoustic models. In the latter service, as the user is allowed to freely formulate her/his request, a word spotting system was implemented which needed improved filler models.

2.1.1 ACI service

In the **ACI** service most of the information (e.g., menu choices, confirmations, dates, credit card numbers, etc.) has to be provided in DTMF mode, with the exception of the car plate, which is alphanumeric. At first, users are invited to freely pronounce the numbers and letters of their car plates and, in case of recognition errors, they are asked to use city names. After three consecutive errors, users are transferred to a human agent. Some examples of car plates (plate patterns are different for cars, motorbikes, etc.) are:

"Milano one B as Bologna two three four five",
"M I one two three four five B",
"Trento one two three four five",
"A B one two three C D",
"A as Ancona B as Bologna one hundred twenty three C as Catania D",
etc.

In addition, speech recognition grammars can take into account some expressions such as: *"hello"*, *"good morning"*, *"the plate number is ..."*, *"my plate is ..."*, etc. Currently this service can automatically manage about 50% of the incoming calls.

	speech	#sentences	#words	source
Baseline (BL)	37 h	51410	187031	various
ACI train (TS)	18 h, 14 min	10472	61967	field
ACI dev (DS)	13 min	146	1035	field
ACI test	46 min	400	2892	field
Unicredit	40 min	887	2486	trials

Table 1

Sizes of the speech databases used in the paper.

With this service it was possible to initially collect about 4 hours of telephone calls. This material was automatically transcribed and used to refine speech recognition grammars, particularly for improving the application domain coverage. Successively, quite a large number of field interactions (about 20 hours) was recorded and divided into training, development and test sets as reported in Table 1.

Table 1 (first row) reports also the data related to the database used to train our standard telephone acoustic models. It stems from various sources and includes about 37 hours of speech. It contains (Falavigna and Gretter, 1997) phonetically rich sentences, confirmation utterances, alpha-letters, digits and the telephonic part of a large broadcast news database (Federico, 2000). About 16 hours of the whole set are the telephonic filtered version of 16 kHz speech acquired in a quiet environment. For this database, manual transcriptions of all utterances are available.

2.1.2 Unicredit service

The **Unicredit** voice portal allows users to obtain information on bank products and services, retrieve investment funds quotations, learn the balance of their current accounts, ask for a fax containing the list of recent transactions, check the status of their orders, manage their access credentials and stop their credit cards. For this service, examples of possible user utterances are:

"I want a fax with the investment funds quotation",
"I need the balance of my account",
"Tell me the quotation of Tiscali",
"I want to listen to the list of my last 10 transactions",
etc.

Initially, only a small set of sentences (887 in total, about 40 minutes of speech) was collected from trials carried out by 20 bank employees. This material, also given in Table 1, was used to build and refine grammars for different word spotting solutions, using the k-fold cross validation technique described in section 4.2. Since a satisfactory performance level was reached before the service went in the production phase, only few refinements were performed on the field.

2.2 Definition of the voice interface

In both services described above, the design of the Voice User Interface (VUI) has required not only traditional system analysis and software development skills but also the specialized skills of linguists, speech scientists, human factors specialists and business process analysts. Relying on these skills, we ap-

proached the task of producing both speech applications from the point of view of the traditional software development life cycle. After an accurate analysis of the requirements, we defined interactive dialogs taking into account user logic, rather than application design logic. In the implementation phase, we created user friendly messages, to be pronounced by the system through a TTS, with regards to the call flow. The application test was divided in two parts: user testing and pilot assessment. For user testing, we asked a small, selected group of subjects (about 20 people) to use the service: the behavior and feedback gathered from these subjects provided key information for validating the accuracy of the dialog. For pilot assessment, we tested the application on a broader basis in a real-life situation. This phase is essential to guarantee a good level of user satisfaction. Statistical analysis of the collected audio traces (see section 4) provides suggestions for final tuning before the production phase. After final deployment, the maintenance of the application completes the project cycle. Since even a small modification may significantly affect the usability of the service, any modification will have to be validated in order to assure the initial "*ear and feel*" design to users. The main difficulties we met during the deployment of the services concerned the usability test, i.e. the process by which we evaluate the effectiveness of the VUI by validating the clarity of voice prompts and the domain coverage provided by speech recognition grammars. The usability test proved to be time consuming and expensive, since the application has to match the requirements, especially when the result implies to change prompts, grammars, or dialog strategies.

Another critical aspect of the VUI design concerns the capability to take into account "user experience". Actually, experienced users needs less information (e.g., more concise voice prompts), from the service, than new or infrequent

users while, at the same time, they learn to utter sentences that are covered by speech recognition grammars. As a consequence, the adopted dialog strategies should adapt to the level of user experience, otherwise the dialog success rate could be worse than expected, even after several grammar refinement steps. We especially observed this fact immediately after the release of the service.

3 Acoustic and language models

Context Independent (CI) phone models were used in both services mentioned above, while in the **ACI** service Context Dependent (CD) models were also exploited and evaluated. The language models consist of a combination of both stochastic and regular grammars (Falavigna et al., 2000).

3.1 Acoustic models

Acoustic phone models are defined by means of three-state left-to-right Hidden Markov Models (HMMs) (Rabiner, 1989), having output densities defined by mixtures of Gaussian densities with diagonal covariance matrices.

The observation vectors are obtained through short term spectral analysis, carried out at a frame rate of 10 ms.

For each 20 ms speech frame, resulting after the application of preemphasis and Hamming window, 12 linear prediction cepstral coefficients, plus the frame log-energy, are evaluated. In addition, to compensate for the distortion introduced by the telephone channel, cepstral mean subtraction (Furui, 1981) is performed over a sliding window of 1 sec (100 frames). At the same

time, the maximum log-energy value, inside the 1 *sec* normalization window, is subtracted to the log-energy of the current analyzed frame. Finally, the first and second order time derivatives of both cepstral and energy parameters are evaluated and inserted into a 39-component feature vector. Time derivatives are estimated applying first and second order regressions to windows formed by 5 and 7 frames, respectively.

All the telephone utterances contain head and tail silences having durations of 200 *ms* and 1 *sec*, respectively. This is due to the behavior of the endpoint detection algorithm applied to the telephone signal before being sent to the recognizer. The algorithm (Orlandi et al., 2003b), which make use of a dynamic energy threshold, allows the system to add a programmable number of signal frames before and after the detected speech boundaries.

Context Independent (CI) models correspond to a "generalized" set of phone units (92 in total), which includes word dependent phone models for digits and confirmations; specific models for background noise, breath and hesitations are also employed.

Context Dependent (CD) models correspond to a set of word independent triphone units. Initially, a set of single Gaussian models is trained for all triphones having a number of occurrences in the training database exceeding a certain threshold. Experiments carried out on the transcription of an Italian broadcast news corpus (Federico, 2000) suggested that 5 was an optimal value for this threshold. If the number of occurrences of a triphone is lower than the threshold, then that triphone model is substituted by a "back-off" model, which can be either a diphone or a generalized phone model, selected according to the same criterion of exhibiting a number of occurrences greater than 5 in

the training database. Then, an "optimal" tying among the triphone states is found using a Phonetic binary Decision Tree (PDT) (Young et al., 1994).

The PDT construction algorithm is similar to the one reported in (Odell, 1995; Young et al., 1994). The leaves of the PDT define clusters that contain triphone states (tied states) sharing a common output mixture density. To estimate the number of training frames that align with the PDT nodes, we use the Baum-Welch density counters of an initial untied HMM set. Hence, to achieve robust estimates of the triphone densities we only generate PDT nodes whose frame occupancy exceeds a given threshold (the latter will be referred henceforth as minimum occupancy threshold). Note that during the PDT construction we do not perform any realignment with the training data.

In Figure 1 Word Accuracy (WA) is plotted as a function of the total number of tied states in the CD model sets used for recognizing the **ACI** development set (see Table 1 for its definition). In the Figure, the different curves correspond to sets of triphones having the same given number of output densities.

To vary the numbers of tied states in the CD models of Figure 1, several triphone sets have been initialized using PDTs estimated with different minimum occupancy thresholds. Then, a number of Baum-Welch learning iterations has been carried out for each set of triphones and, before starting a learning iteration, the densities with the largest accumulated Baum-Welch counters have been split (i.e. two densities are artificially derived from each given density) till reaching the desired overall number of output densities.

To train the models of Figure 1 we used **BL** ∪ **TS**, i.e. the union of the baseline task independent database **BL**, defined in Table 1, and of the task specific training database **TS**, also defined in Table 1.
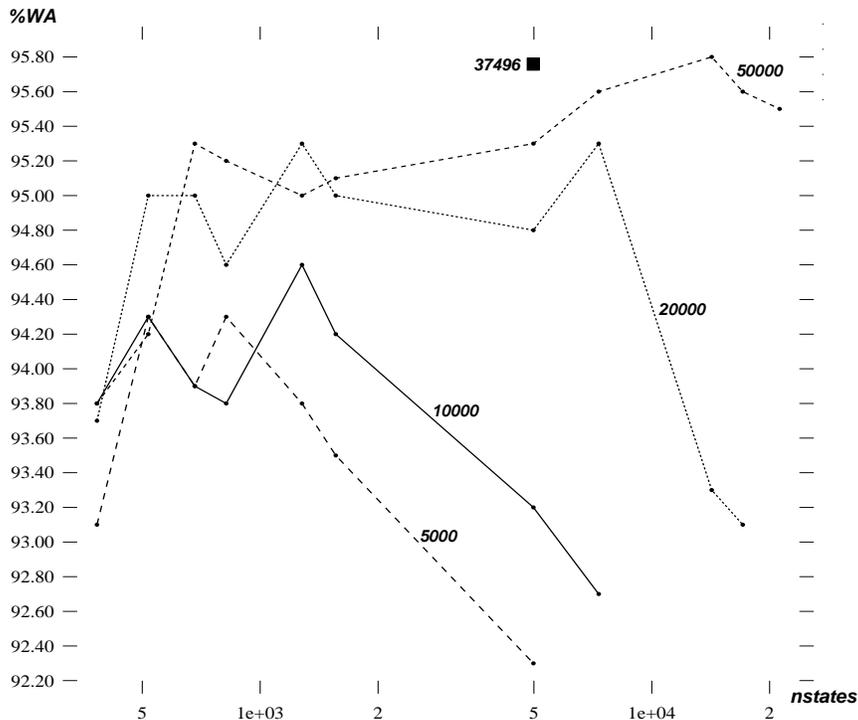
Fig. 1. *Performance obtained with different sets of CD models on the **ACI** development speech database (**DS**). Word Accuracy (vertical axis) is plotted against total number of tied states (horizontal axis, on a log scale) for different numbers of output probability densities. The black square corresponds to the CD model set used in the experiments reported in section 4.*

As one can expect, ASR performance depends on both the total number of HMM states and the number of densities and, in general, over a certain value of HMM states (around 1500 in the Figure), WA increases as the total number of trainable parameters (i.e. approximately the number of Gaussian parameters) increases. However, models exhibiting higher numbers of densities require more states to become effective: in fact, a reduced number of states increases the level of density tying in the models. Furthermore, we observe a generalized performance decrease at increasing number of tied states. This indicates that a minimum number of densities is necessary to achieve accurate modeling of acoustic spaces characterized by high state resolutions (note that the number

15

of output densities, in the various model sets, must be necessarily greater than the number of tied states).

To design the acoustic models used in the experiments reported in section 4, we took advantage of experience gained from an Italian broadcast news transcription task afforded in the past (Federico, 2000). We selected 9541 triphone models, each occurring more than 5 times in the training database. Then, we determined 4986 tied states using a PDT with minimum occupancy threshold equal to 1000. The latter value is considered sufficient to robustly estimate 8 Gaussian densities per state, starting from a single Gaussian mixture and applying 3 successive splits. The number of output densities at the end of the training phase was 37496 (note that output densities having a total variance below a given threshold were pruned).

With this last set of CD models a WA of 95.8% (this performance is also shown, as a black square, in Figure 1) was obtained on the development set of the **ACI** task. Note that to draw the curves of Figure 1, where a predefined number of densities was forced, a different allocation method was used. On the same task the CI model set (formed by 92 models, 427 states and 3724 output densities), trained on $\mathbf{BL} \cup \mathbf{TS}$, gives 92.5% WA.

Models	#mod	#distrs	MBytes	secs
CI	92	3724	2.4	104
CD	9541	37496	27	996

Table 2

Memory and computation requirements for CI and CD models. The evaluation was carried out on the **ACI** *development set.*

It is worth noticing the additional memory and computation requirements of the CD models with respect to the CI ones. Table 2 gives the number of models (**#mod**), number of densities (**#distrs**), the corresponding memory occupation (in megabytes) and computation time (in seconds) for decoding the whole development set of the **ACI** task.

As can be seen from Table 2, there is about one order of magnitude between CI and CD models complexities. For a given dimension of the telephone platform, basically depending on the number of telephone lines, the employment of CD models requires augmenting the the computational resources by a factor of almost 10. Figure 1 also suggests that the choice of CD model size is an engineering tradeoff between computational expense and recognition accuracy.

3.2   Acoustic training exploiting task specific data

We investigated the possibility to adapt, in a completely unsupervised way (or with a small manual effort), the general purpose CI models to the specific task. We developed an automatic selection algorithm in order to find an "optimal" (in a way to be specified) subset of task specific data to be added to the baseline database with the purpose of training models. The chosen optimality criterion was to minimize the Word Error Rate (WER); alternatively, the Sentence Error Rate (SER) could be used. The core of the algorithm lies in the use of rejection grammars; alternatively, confidence measures (Gretter and Riccardi, 2001) could be exploited. A recognition step is carried out on a set of task specific data by putting a rejection network in parallel with the recognition grammar. This allows us to discard a number, NS, of speech segments, $\mathbf{O}_s$, $1 \leq s \leq NS$, exhibiting a probability ratio value $\frac{P[rej|\mathbf{O}_s]}{P[rec|\mathbf{O}_s]} > 1$, where rej

and *rec* are rejection and recognition grammars, respectively. Note that the rate of the discarded segments can be controlled by varying the value of the language model probability assigned to the rejection network itself (the higher this value, the higher the number of discarded segments). Then, a training phase is performed using the retained material and the WER is evaluated on a separate development set. This process can be iterated as explained below.

Let us define the following quantities: **BL** the baseline (task independent) training set, $\mathcal{M}_{\mathbf{BL}}$ the baseline acoustic model set, **TS** (see Table 1 for its definition) the initial task specific training set, **DS** a task specific development set, $\mathcal{M}_i$ the acoustic model set used at i^{th} step, $\mathbf{TS}_i$ the task specific training set at step i, $\mathbf{TR}_i$ the speech data rejected at step i and $WER[\mathcal{M}_i, \mathbf{DS}]$ the word error rate obtained on **DS** using acoustic models $\mathcal{M}_i$. The **completely unsupervised** training algorithm proceeds according to the following steps.

(1) `set` $i = 0$; `set` $\mathcal{M}_i = \mathcal{M}_{BL}$; `set` $\mathbf{TS}_i = \mathbf{TS}$; `set` $\mathbf{TR}_i = \oslash$;

(2) `evaluate` $WER[\mathcal{M}_i, \mathbf{DS}]$;

(3) `transcribe` $\mathbf{TS}_i$ `using` $\mathcal{M}_i$;

(4) `train` $\mathcal{M}_{i+1}$ `using` $\mathbf{BL} \cup \mathbf{TS}_i$;

(5) `evaluate` $WER[\mathcal{M}_{i+1}, \mathbf{DS}]$;

(6) `increase the probability of the ''rejection'' network`
    `in the language model;`

(7) `put each sentence of` $\mathbf{TS}_i$ `either in` $\mathbf{TS}_{i+1}$ `or in` $\mathbf{TR}_{i+1}$,
    `depending on its probability ratio;`

(8) `if` $WER[\mathcal{M}_{i+1}, \mathbf{DS}] > WER[\mathcal{M}_i, \mathbf{DS}]$
    `then return with` $\mathcal{M}_i, \mathbf{TS}_i$;

```
else set i = i + 1; goto (3).
```

Note that determining the rejection network probability is somewhat task dependent, although not critical.

The other method we investigated for model training requires manual transcription of a set of data, **TR**, derived in some "useful" way (see section 4.1.2 for the details) from the rejected sets $\mathbf{TR}_i, 1 \leq i \leq I$, defined above, where I represents the total number of iterations run. Then, a training step is performed over the set $\mathbf{BL} \cup \mathbf{TR}$, using the available manual transcriptions. This technique has been called **partly supervised** training.

Finally, the **completely supervised** method requires manual transcription of the whole set **TS** and training over $\mathbf{BL} \cup \mathbf{TS}$ using all the available manual transcriptions.

3.3 Language models

The language models adopted in the services are based on Recurrent Transition Networks (RTNs) (Brugnara and Federico, 1997; Demori, 1998). RTNs are Finite State Networks (FSNs) whose transitions can be either terminal symbols or links (also recursive) to other FSNs, enriched with syntactic or semantic labels. In terms of expressive power, RTNs are equivalent to context free grammars. Since the decoding step of a speech utterance can backtrack both the FSNs, with their associated labels, and the words along the best path in the search space, the recognized string consists of a mix of words and syntactic (or semantic) labels. Different types of RTNs are available, exhibiting increasing complexity, for example lists of words or phrases, regular

19

expressions and stochastic language models (bigrams or trigrams). Even filler models (either phone or word loops), used in word spotting applications, can be considered RTNs.

As an example, the following "regular expressions" define two RTNs that allow the system to recognize car plates in two different formats, called "`old_plate`" and "`new_plate`".

```
alpha     a|b|c|...|x|y|z
digit     0|1|2|...|7|8|9
city      ancona|bologna|...|milano|...|roma|...|venezia
old_plate city:CITY (alpha:A1|digit:D1) (alpha:A2|digit:D2)
          digit:D3 digit:D4 digit:D5 (alpha:A6|digit:D6)
new_plate alpha:A1 alpha:A2 digit:D1 digit:D2 digit:D3
          alpha:A3 alpha:A4
```

In the definitions above, symbol "|" indicates alternatives; semantic labels, when given, follow colons. RTNs defined in the example above can, in turn, be used to form the network shown in Figure 2, which was employed in the ACI service for recognizing the plate sentences.

A possible output for a given utterance (e.g., "`my car plate is a b 1 2 3 c d`") recognized with the grammar **ACI** in Figure 2 is:

```
my car plate is (NPLATE( (A1( a )A1) (A2( b )A2) (D1( 1 )D1)
(D2( 2 )D2) (D3( 3 )D3) (A3( c )A3) (A4( d )A4) )NPLATE)
```

The example above can be interpreted as a parse tree, where semantic labels are surrounded by pairs of round brackets. Words outside semantic labels do
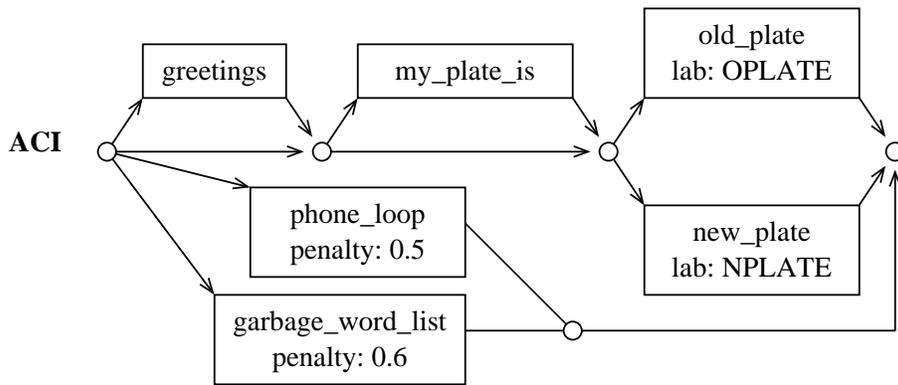
Fig. 2. *A composite RTN, in graphical form. Boxes represent links to other RTNs, semantic labels follow the symbol* `lab:`*, penalties allow the system to control the rejection rate. Where not explicitly indicated, the value of the penalty is zero.*

not carry useful information and can be discarded.

The rejection of a sentence occurs when the best path crosses one of the RTNs indicated in Figure 2 as *phone_loop* or *garbage_word_list*; the rejection rate can be controlled by modifying the values of the penalties associated to these last two networks.

3.3.1 ACI language model

In the **ACI** service a set of 118 hand crafted RTNs was employed, covering all possible plate formats (cars, motorbikes, etc.) and allowing users to express alpha-letters in different ways (the whole vocabulary of the **ACI** service consists of 287 words). For instance, the alpha-letters "a" and "b" can be recognized through the two RTNs shown in Figure 3. First, these grammars were built using a-priori knowledge, then they were refined by looking at the first 4 hours of speech collected with the service (see subsection 2.1.1). Development, test and training data were collected later.
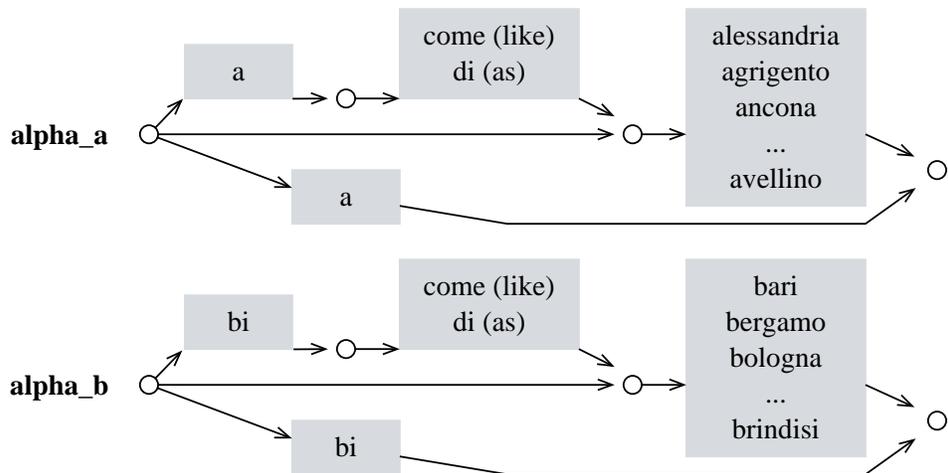
Fig. 3. *RTNs used to recognize the alpha-letters "a" and "b". Grey boxes contain terminal words.*

3.3.2 Unicredit language model

The **Unicredit** task is basically a phrase spotting task; the number of phrases to spot, according to the used RTNs, is 273. Of these, 145 are composed by a single word, 88 by a couple of words and the remaining 40 by 3 to 5 words. The total number of different words to spot is 270, while the total number of words of the application, referred in the final grammar, is 861. A certain number of basic RTNs were defined for handling the semantic information and, in addition, some filler grammars and several main grammars were designed for a total of 104 RTNs. All the main grammars share the same topological structure: a loop of alternatives formed by semantic RTNs and filler RTNs. Filler grammars were designed in order to minimize the WER over the sentences collected in the initial development phase (see Table 1). Due to the small amount of available data, a k-fold cross validation technique for WER evaluation was used. Note that the "top level" main grammar may be either stochastic or hand crafted, and links the other semantic grammars as previously seen. For instance, sentences like:

```
 I  want to  listen to  the list  of  my last

   transactions tell me the quotation of Tiscali
```

are processed by the recognizer working in text mode to obtain:

```
 I want (VOICE( to listen to )VOICE) the list of my last

 (TRANS( transactions )TRANS)

 (VOICE( tell me )VOICE) the quotation of (QUOT( Tiscali )QUOT)
```

which easily becomes:

```
 I want VOICE the list of my last TRANS

 VOICE the quotation for QUOT
```

which is finally used to train a class trigram language model (Brown et al., 1992), where `VOICE`, `TRANS` and `QUOT` are not words but links to RTNs. Note that terminal symbols (i.e. words) not surrounded by semantic labels only contribute to the estimation of filler grammars.

4   Experiments and results

The experiments reported in this section were conducted on both the **ACI** and **Unicredit** tasks. In both cases the CI acoustic model set, trained on the baseline task independent database **BL**, was used as starting point. On the **ACI** task the baseline CI models were trained over baseline plus task specific data according to the procedures described in section 3.2; on the **Unicredit** task only the language model effects were investigated.

4.1 ACI task

On this task, the completely unsupervised and partly supervised training procedures were applied to the baseline CI model set. Successively, comparisons were drawn with corresponding sets of CD models trained on the resulting task-specific database partitions.

4.1.1 Completely unsupervised training

The completely unsupervised training algorithm, described in section 3.2, was applied to the set **TS** defined in Table 1. The database partition resulting at the end of the process is shown in Figure 4. Four iterations of the unsupervised training procedure were run (the stop condition of the algorithm was determined using the **ACI** development set, also defined in Table 1), giving subsets TS_1 (16 h), TS_2 (13.7 h), TS_3 (11.2 h) and TS_4 (9.7 h).
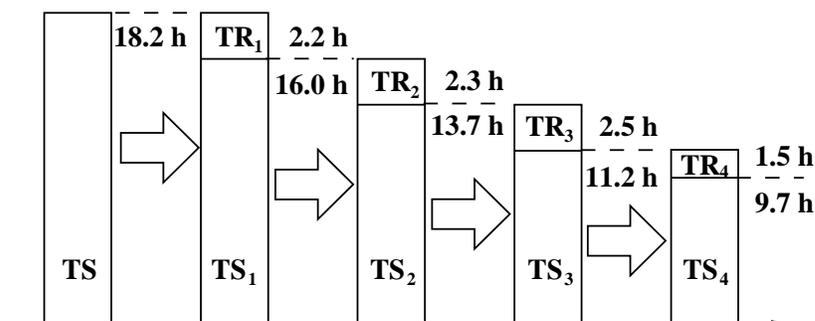

Fig. 4. *Automatic selection of task specific subsets.*

The baseline CI model set was successively trained using **BL** plus the task specific databases **TS**, TS_1, …, TS_4. Results obtained on both the **ACI** development and test sets are given in Table 3.

24

Training	Dev:WER	Dev:SER	Test:WER	Test:SER
BL	11.0%	45.2%	14.7%	45.8%
$\textbf{BL} \cup \textbf{TS}(18.2h)$	9.6%	39.7%	13.3%	43.5%
$\textbf{BL} \cup \textbf{TS}_1(16.0h)$	9.5%	37.0%	11.5%	38.0%
$\textbf{BL} \cup \textbf{TS}_2(13.7h)$	9.3%	35.6%	**11.3**%	38.0%
$\textbf{BL} \cup \textbf{TS}_3(11.2h)$	**8.8**%	**34.9**%	**11.3**%	**37.3**%
$\textbf{BL} \cup \textbf{TS}_4(9.7h)$	9.0%	**34.9**%	11.6%	38.0%

Table 3

*WER and SER obtained on the **ACI** task using **completely unsupervised** training; best results are given in bold. The context independent HMMs have been used.*

Note that best performance (shown in bold in Table 3) was obtained training over the set $\textbf{BL} \cup \textbf{TS}_3$. Note also the benefit brought by the proposed completely unsupervised training procedure (from 14.7% WER to 11.3% WER, about 20% relative WER reduction).

4.1.2 Partly supervised training

The basic principle underlying the partly supervised training procedure consists in determining a method for selecting from **TS** a "useful" set of speech files to manually transcribe and use for training. The criterion that we adopted for obtaining this subset was to remove, from **TS**, as many sentences (or parts of sentences) as possible that are either correctly recognized or that do not contain alphanumeric sentences. Instead, we want to retain a significant number of alphanumeric sentences that are not correctly recognized. To this purpose

we analyzed a small number of sentences (the first 50 sentences, in chronological order) taken from every subset $\mathbf{TR}_i, 1 \leq i \leq 4$, of Figure 4, according to the following four classes:

- $\mathbf{P+}$: plates correctly recognized;
- $\mathbf{P-}$: plates with at least one recognition error;
- $\mathbf{C}$: speech out of context (non plates);
- $\mathbf{R}$: background noise only.

set	P+	P−	C	R
$\mathbf{TR}_1$	5	21	14	10
$\mathbf{TR}_2$	5	44	1	0
$\mathbf{TR}_3$	13	36	1	0
$\mathbf{TR}_4$	24	26	0	0

Table 4

Classification of 50 sentences belonging to the rejected subsets $\mathbf{TR}_i$.

The distribution of the content of the analyzed sentences, according to the classes defined above, is shown In Table 4. Note that $\mathbf{TR}_4$ contains only alphanumeric sequences, half of which were correctly recognized. $\mathbf{TR}_1$ contains a higher percentage of wrongly recognized plates, as well as a high percentage of "non-plates" sentences. Both $\mathbf{TR}_2$ and $\mathbf{TR}_3$ exhibit high error rate and contain few "non-plates" data, hence they appear to be the most suitable material to manually transcribe for the partly supervised training step. As a consequence, $\mathbf{TR}_2$ and $\mathbf{TR}_3$ were manually transcribed and renamed $\mathbf{TR}_{mt}$. After having trained on $\mathbf{BL} \cup \mathbf{TR}_{mt}$ ($\mathbf{TR}_{mt}$ is formed by 4.8 hours speech) we obtained the results given in the first row of Table 5.

Training	Dev:WER	Dev:SER	Test:WER	Test:SER
$\mathbf{BL} \cup \mathbf{TR}_{mt}(4.8h)$	7.9%	33.6%	9.0%	31.2%
$\mathbf{BL} \cup \mathbf{TR}_{mt} \cup \mathbf{TS}_3$	7.5%	32.9%	9.4%	33.8%

Table 5

*WER and SER obtained on the **ACI** task with **partly supervised** training. The context independent HMMs have been used.*

A comparison of the results given in Table 5 (first row) with those reported in Table 3 (fifth row), related to the completely unsupervised method, clearly shows the effectiveness of the proposed procedure: on the test set WER decreased from 11.3% to 9.0%, with a further relative reduction of about 20%. We also tested the effect of adding the completely unsupervised (automatically transcribed) database $\mathbf{TS}_3$ to $\mathbf{BL} \cup \mathbf{TR}_{mt}$, i.e. we trained on $\mathbf{BL} \cup \mathbf{TR}_{mt} \cup \mathbf{TS_3}$. The resulting performance, which is very close to the the previous one, is shown in the second row of Table 5.

4.1.3 Context independent vs. context dependent models

The database partitions resulting from the application of the completely unsupervised and partly supervised procedures were used to train the set of CD units defined in section 3.1 (Table 2). In addition, a completely supervised training step was carried out for both CI and CD model sets, using $\mathbf{BL} \cup \mathbf{TS}$, where all the manual transcriptions of $\mathbf{TS}$ are available. Performance is given in Table 6. For convenience, in the same Table, performance related to CI models are also shown.

From Table 6 we notice two main results.

Training	Models	Test: WER	Test: SER
completely supervised $(\mathbf{BL} \cup \mathbf{TS})$	**CI**	8.7%	30.5%
partly supervised $(\mathbf{BL} \cup \mathbf{TR}_{mt})$	**CI**	9.0%	31.2%
completely unsupervised $(\mathbf{BL} \cup \mathbf{TS}_3)$	**CI**	11.3%	37.3%
completely supervised $(\mathbf{BL} \cup \mathbf{TS})$	**CD**	5.7%	27.5%
partly supervised $(\mathbf{BL} \cup \mathbf{TR}_{mt})$	**CD**	6.9%	32.2%
completely unsupervised $(\mathbf{BL} \cup \mathbf{TS}_3)$	**CD**	7.7%	37.5%

Table 6

Performance obtained on the **ACI** *test set using both context independent and context dependent HMMs. The three different training procedures are compared.*

The first one refers to the improvement achieved with CD models (5.7%) with respect to CI ones (8.7%): 50% relative WER reduction in case of completely supervised training. This WER decrease is similar to that obtained on different tasks reported in the literature (Odell, 1995; Young et al., 1994; Lamel et al., 2001a).

The second important result shows that the partly supervised training method gives performance close to that furnished by the completely supervised procedure. This is particularly evident for the CI model sets (compare first row of Table 6, 8.7% WER, with second row, 9.0% WER). However, it is worth noticing the much smaller size of the manually transcribed training set $\mathbf{TR}_{mt}$ ($4.8h$), used in partly supervised training, with respect to the size of the whole database **TS** ($18.2h$), used in completely supervised training.

4.1.4 Discussion

When using unsupervised training, three issues have to be considered.

The first one refers to the size of the training material: the more the data, the better the improvement.

The second issue concerns the quality of the transcriptions: if the data contain a low percentage of transcription errors, training will improve acoustic models; but, as this percentage grows, the resulting models will tend to learn errors and finally to diverge, lowering the recognition performance.

The third issue concerns the real efficacy of the information carried by the correctly recognized sentences. Intuitively, the information carried by such sentences is already known by the system. Instead, wrongly recognized sentences - if correctly labeled - carry some new information and hence could be more effective.

The results obtained with completely unsupervised training can be explained following the line of reasoning behind the first two issues above. This technique does indeed allow the system to reach a reasonable compromise between the quantity and the quality of the training data.

In the case of partly supervised training the best compromise can be reached, since only the data carrying "new" information are retained (i.e. the sentences that are not correctly recognized), while training is carried out employing the corresponding correct manual transcriptions. We observe that the relative difference between WER provided by partly supervised and completely supervised training methods is lower for CI models (9.0% vs. 8.7%, in Table 6) than for CD models (6.9% vs. 5.7%); this is probably due to the fact that the quan-

tity of supervised training data (i.e. the size of the database $\mathbf{TR}_{mt}$) is small with respect to the increased number of statistical parameters to estimate for CD models.

Finally, we point out that in some systems implementing completely supervised training, sentences that fail to align against their transcription for a given width of the beam search are discarded. This resembles in some sense the rationale of the first step of our unsupervised training procedure, in which rejected sentences are discarded.

4.2   Unicredit task

As mentioned above, experiments carried out on this task used the baseline CI acoustic models defined in section 3.1.

Users of **Unicredit** service (see section 3.3), instead of being presented a long menu, are invited to freely ask for the desired information.

To extract only the semantically relevant information from input utterances of the **Unicredit** service, either word or phrase spotting is performed. This is done through an "optimal" use of rejection grammars.

Analogously to the **ACI** task, an initial set of sentences (887 in total, about 39.4 minutes of speech) was collected from trials carried out by 20 bank employees. All the test sentences were manually transcribed at both the orthographic and semantic levels. For instance, the sentence "`I want to listen to the list of last transactions`" contains the two semantic tokens: (`VOICE( to listen to )VOICE`) and (`TRANS( transactions )TRANS`). The remaining

words (i.e. "I want" and "the list of last") must be rejected.

The whole test set contains 2486 words (2.8 words per sentence, on average) and 769 semantic units. Before starting the acquisitions, a grammar was designed using only a-priori domain knowledge. This initial grammar consisted of a loop of all the phrases to spot, mixed with a generic rejection grammar. We noticed that 265 of the test sentences (29.9%) were covered by the grammar itself; the remaining 622 sentences (70.1%) were out of the grammar coverage and, therefore, need to be totally or partially rejected. Consequently, we defined several rejection grammars to be used in opposition with the phrases to spot, with the purpose of balancing the number of false alarms and recognition errors on the test set. Furthermore, since the test database is quite small, a k-fold cross validation technique, using $k = 4$, was adopted, consisting in splitting the test sentences into 4 subsets of equal size. Three subsets were used as development set for estimating grammar parameters (e.g., garbage word lists, rejection probabilities, stochastic language models), and the other (test set) for measuring performance. In successive steps, each subset became in turn the test set while the others formed the development set; in this way, the overall performance can be evaluated exploiting the whole database.

The resulting performance, expressed in terms of semantic errors, is shown in Table 7. In the Table, **#SER** and **#WER** represent sentence error rate and word error rate; **#Err** is the total number of semantic errors, **#D+#I+#S** give the number of semantic deletion, insertion and substitution errors, respectively.

The baseline rejection grammar (first row in Table 7) simply consists of a phone loop. To reduce the number of insertion errors (223) provided by this

Rejection grammar	SER	WER	#Err	#D+#I+#S
phone loop	27.4%	36.7%	282	(34+223+25)
tuned phone loop	20.6%	26.5%	204	(57+124+23)
garbage lexicon	13.7%	17.4%	134	(50+63+21)
garbage unigrams	12.9%	16.2%	125	(46+58+21)
bigrams	9.1%	12.9%	99	(27+54+18)
trigrams	8.0%	11.1%	85	(27+42+16)

Table 7

Performance obtained on the **Unicredit** *test set using different rejection grammars. The baseline context independent HMMs have been used.*

grammar, we estimated an optimal penalty to be added to the overall log-likelihood, evaluated during the Viterbi search, before entering the rejection network. As a result, WER decreased from 36.7% to 26.5% (second row in Table 7). To further improve system performance, we built a garbage lexicon containing all the words in the development sets not belonging to valid sentences. This grammar was again put in competition with the other two grammars, enabling the recognition of any combination of valid phrases, rejected phones and garbage words. This reduced WER to 17.4% (third row in Table 7). A further - albeit small - WER improvement was achieved (16.2%, fourth row in Table 7) by adding unigram probabilities to the garbage lexicon.

In the last two experiments, we processed the sentences in the development sets in order to estimate a bigram / trigram language model, as explained in section 3.3.2. These training data allow us building stochastic grammars

that include both words and semantic classes. Due to the limited size of the training material, the language model probabilities were smoothed using the shift-beta method without threshold, i.e. all bigrams / trigrams were retained (Bertoldi and Federico, 2004). Despite the very small training data, best WER performance (12.9% and 11.1% for bigrams and trigrams, respectively) was obtained with this approach.

Note that in terms of relative WER, the most notable improvements come from the use of the garbage lexicon (34% relative decrease with respect to tuned phone loop grammar) and the stochastic language models (bigrams give 20% relative reduction with respect to unigrams).

5 Conclusions

In this paper, we have described some of the problems encountered during the design and development phases of two telephone applications: **ACI**, to pay the road tax automatically, and **Unicredit**, to access bank information.

We have shown that the **ACI** service can be significantly improved by means of acoustic model retraining. For this service we have demonstrated the effectiveness of both a totally unsupervised acoustic training procedure and a partly supervised one, based on an "optimal" selection of the speech material to be transcribed manually. We have also measured the performance improvement reachable with context dependent phone models; these last ones require much larger computation and memory resources - we measured an increase of about one order of magnitude - with respect to context independent phone models.

On the **Unicredit** service, which is mainly a word spotting task, we have evaluated the effectiveness of different rejection networks to model the language to discard. We have shown that a class trigram language model allows WER to be reduced from 36.7% to 11.1% with respect to a phone loop based network.

Most of the development effort was spent for usability test, which proved to be both time consuming and expensive. Human factors, statistical analysis, evaluation of dialog success rates and tuning on field data strongly affect the overall costs of services.

6   Acknowledgments

References

Bertoldi, N., Federico, M., 2004. Broadcast news LM adaptation over time. Computer Speech and Language 18 (1), 417–435.

Brown, P. F., Pietra, V. J. D., deSouza, P. V., Lai, J. C., Mercer, R. L., 1992. Class-based n-gram models of natural language. Computational Linguistics 18 (4), 467–479.

Brugnara, F., Federico, M., 1997. Dynamic language models for interactive speech analysis. In: Proc. of Eurospeech. Rhodes, Greece, pp. 2751–2754.

Demori, R., 1998. Spoken Dialogues with Computers. Academic Press, London.

Facco, A., Falavigna, D., Gretter, R., Viganò, M., October 2004. On the development of telephone applications: Some practical issues and evaluation. In: Proc. ICSLP. Jeju, Korea, pp. 2625–2628.

Falavigna, D., Gretter, R., 1997. On field experiments of continuous digit recognition over the telephone network. In: Proc. of Eurospeech. Rhodes, Greece, pp. 1827–1830.

Falavigna, D., Gretter, R., 1998. Telephone speech recognition applications at IRST. In: Proc. IEEE Workshop on Interactive Voice Technology for Telecommunications Applications. Turin, Italy, pp. 27–30.

Falavigna, D., Gretter, R., Orlandi, M., 2000. Mixed language model for a dialogue system over the telephone. In: Proc. ICSLP. Beijing, China, pp. 585–588.

Federico, M., 2000. A system for the retrieval of Italian broadcast news. Speech Communication 1 (32), 37–47.

Freund, Y., Schapire, R., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55 (1), 119–139.

Furui, S., 1981. Cepstrum analysis technique for automatic speaker verification. IEEE Trans. on Acoustics Speech and Signal Processing 29 (1), 245–272.

Gorin, A., Riccardi, G., Wright, J., 1997. How may I help you ? Speech Communication 23, 113–127.

Graff, D., 2002. An overview of broadcast news corpora. Speech Communication 37, 15–26.

Gretter, R., Riccardi, G., May 2001. On-line learning of language models with word error probability distributions. In: Proc. of ICASSP. Salt Lake City, US, pp. 557–560.

Harald, A., Schroer, O., February 1998. An overview of the Philips dialog system. In: Proc. of the DARPA Broadcast News Transcription and Understanding Workshop. Lansdowne, VA, US.

He, Y., Young, S., 2003. Data-driven spoken language understanding system. In: Proc. of Automatic Speech Recognition and Understanding Workshop. St. Thomas, US, pp. 583–588.

Kamm, T., Meyer, G., December 2001. Automatic selection of transcribed training material. In: Proc. of ASRU. Madonna di Campiglio, Italy.

Kamm, T., Meyer, G., October 2004. Robustness aspects of active learning for acoustic modeling. In: Proc. of ICSLP. Jeju Island, Korea, pp. 1973–1976.

Lamel, L., Gauvain, J., Adda, G., September 2000. Lightly supervised acoustic model training. In: Proc. of ISCA ITRW ASR2000. Paris, pp. 150–155.

Lamel, L., Gauvain, J., Adda, G., May 2001a. Investigating lightly supervised acoustic model training. In: Proc. of ICASSP. Salt Lake City, US.

Lamel, L., Gauvain, J., Adda, G., 2002. Lightly supervised and unsupervised acoustic model training. Computer Speech and Language 16 (1), 115–129.

Lamel, L., Gauvain, J., Lefevre, F., May 2001b. Towards task-independent speech recognition. In: Proc. of ICASSP. Salt Lake City, US.

Levin, E., Pieraccini, R., 1995. CHRONUS, the next generation. In: Proc. ARPA Spoken Language Systems, Technology Workshop. Austin, TX, US.

Levin, E., Pieraccini, R., Eckert, W., 2000. A stochastic model of human-machine interaction for learning dialog strategies. IEEE Trans. on Speech and Audio Processing 8 (1), 11–23.

Odell, J., 1995. The Use of Context in Large Vocabulary Speech Recognition. Ph. D. Thesis, Cambridge University, UK.

Orlandi, M., Culy, M., Franco, H., 2003a. Using dialog corrections to improve speech recognition. In: Proc. of ISCA Workshop on Error Handling in Spo-

ken Dialog Systems. Geneve, Switzerland, pp. 47–51.

Orlandi, M., Santarelli, A., Falavigna, D., September 2003b. Maximum likelihood endpoint detection with time-domain features. In: Proc. Eurospeech. Geneve, Switzerland, pp. 1757–1760.

Pallet, D., Fiscus, J., Garofolo, J., Martin, A., Przybocki, M., March 1999. 1998 broadcast news benchmark test results. In: Proc. NIST DARPA Broadcast News Workshop. Washington, US, pp. 1–5.

Puterman, M., 1994. Markov Decision Processes. Wiley ed., New York.

Rabiner, L. R., 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of IEEE Trans. on Acoustics, Speech and Signal Processing 77 (2), 267–295.

Ward, W., Issar, S., 1995. The CMU ATIS system. In: Proc. ARPA Spoken Language Systems, Technology Workshop. Austin, TX, US.

Woodland, P., 2002. The development of the HTK broadcast news transcription system: An overview. Speech Communication 80, 2295–2305.

Young, S., 2002. Talking to machines (statistically speaking). In: Proc. ICSLP. Denver, CO, US, pp. 9–16.

Young, S., Chase, L., 1998. Speech recognition evaluation: a review of the U.S. CSR and LVCSR programmes. Computer Speech and Language 6 (4), 263–279.

Young, S., Odell, J., Woodland, P., 1994. Tree-based state tying for high accuracy acoustic modelling. In: Proc. of Human Language Technology Workshop. Morgan Kaufmann, Plainsboro, US, pp. 307–312.

Zue, V., et al., January 2000. Jupiter: A telephone-based conversational interface for weather information. IEEE Trans. on Speech and Audio Processing 8 (1), 85–96.