ELSEVIER

Speech Communication 34 (2001) 195–212

SPEECH COMMUNICATION

www.elsevier.nl/locate/specom

Robust numeric recognition in spoken language dialogue

Mazin Rahim *, Giuseppe Riccardi, Lawrence Saul, Jerry Wright, Bruce Buntschuh, Allen Gorin

AT&T Labs – Research, Room E105, 180 Park Avenue, Florham Park, NJ 07932 USA

Abstract

This paper addresses the problem of automatic numeric recognition and understanding in spoken language dialogue. We show that accurate numeric understanding in fluent unconstrained speech demands maintaining robustness at several different levels of system design, including acoustic, language, understanding and dialogue. We describe a robust system for numeric recognition and present algorithms for feature extraction, acoustic and language modeling, discriminative training, utterance verification and numeric understanding and validation. Experimental results from a field-trial of a spoken dialogue system are presented that include customers' responses to credit card and telephone number requests.

Keywords: Robustness; Spoken dialogue system; Speech recognition; Utterance verification; Discriminative training; Understanding; Language modeling; Numeric recognition; Digits

1. Introduction

Interactive spoken dialogue systems can play a significant role in automating a variety of services including customer care and information access (Lamel et al., 1999; Rahim et al., 1999; Riccardi and Gorin, 1999; Ramaswamy et al., 1999; Os et al., 1999). The success of these systems depends largely on their ability to accurately recognize and understand fluent unconstrained speech. The premise is that users would be able to engage with these systems in a natural open dialogue with minimal human assistance.

In this paper, we describe a class of applications for spoken dialogue systems that involve the use of a *numeric language*. This language includes a set of phrases that form the basis for recognizing and understanding credit card and telephone numbers, zip codes, social security numbers, etc. This language forms an integral part of a field-trial study of AT&T customers responding to the open-ended prompt "How May I Help You?". It consists of several distinct phrase classes such as *digits, natural numbers, alphabets, restarts, city/country name* and other miscellaneous phrases. The numeric language can be considered as a set of "salient" phrases that are relevant to the task of number recognition and which may be identified by exploiting the mapping from unconstrained language into machine action (Gorin, 1995; Wright et al., 1997, 1998).

An essential ingredient for facilitating natural interaction in spoken dialogue systems is maintaining system *robustness*. Although it is commonly considered as a mismatch problem between the acoustic model and the test data (Sankar and Lee, 1996; Rahim and Juang, 1996), robustness in spoken dialogue systems extends to several other dimensions:

* Corresponding author.
E-mail address: mazin@research.att.com (M. Rahim).
Web.: http://www.research.att.com/info/mazin

0167-6393/01/$ - see front matter
PII: S0167-6393(00)00054-6

- *Robust acoustic modeling.* Variations in the acoustic characteristics of the signal due to extraneous conditions (such as background noise, reverberation, channel distortion, etc.) should cause little or no degradation in the performance of the recognizer.
- *Robust language modeling.* Users should be able to express themselves naturally and freely without being constrained by a highly structured language model.
- *Robust language understanding.* The presence of disfluencies (such as "ah", "mm", etc.) and recognition errors should have no or little impact on the behavior of the system.
- *Robust dialogue strategy.* The dialogue manager should guide both novice and skilled users through the application seamlessly and intelligently.

Maintaining robustness at these various levels is the key to the success of spoken dialogue systems in the telecommunication environment.

In this paper, we consider the problem of numeric language recognition in spoken dialogue systems as a large-vocabulary continuous speech recognition task (Rahim, 1999a). We address the technical challenges in developing a robust, accurate and real-time recognition system. We present algorithms for improved robustness at the acoustic, language and understanding levels of the system. We demonstrate that our system for numeric recognition provides robustness to a wide variety of inputs and environmental conditions.

The organization of this paper is as follows. Section 2 describes the challenges involved when dealing with fluent unconstrained speech, and provides evidence that demonstrates the need for utilizing a numeric language as opposed to merely digits. Section 3 provides a characterization of our experimental database for numeric recognition. In Section 4, we describe the various modules of the proposed system, including feature extraction, acoustic modeling and training, language modeling, utterance verification and numeric understanding and validation. The performance of these various modules are presented in the experimental results of Section 5. Finally, a summary and a discussion are provided in Section 6.

2. From digits to numeric recognition

Connected digits play a vital role in many applications of speech recognition over the telephone. Digits are the basis for credit card and account number validation, phone dialing, menu navigation, etc.

Progress in connected digits recognition has been remarkable over the past decade (Cardin et al., 1993; Buhrke et al., 1994). Fig. 1 summarizes the performance of state-of-the-art digit recognizers on a variety of databases that range from high-quality read speech to conversational spontaneous speech (Chou et al., 1995; Buhrke et al., 1994). These results reflect the digit error rate when using a free digit grammar and with no rejection. [1] Under carefully-monitored laboratory conditions, such as the Texas Instrument (TI) database, it is shown that recognizers can generally achieve less than 0.3% digit error rate (Cardin et al., 1993) (see Fig. 1 – "READ"). Dealing with telephone speech adds new difficulties to this problem. Variations in the spectral characteristics due to different channel conditions, speaker populations, background noise and transducer equipment can cause a significant degradation in recognition performance. Nevertheless, through advances in acoustic modeling, discriminative training and robustness, many recognition systems today can operate at 1–2% digit error rate (Chou et al., 1995; Rahim et al., 1996). This is illustrated in Fig. 1 ("FLUENT") for a variety of in-house databases that have been collected over the telephone, including Teletravel (TELE), Voice Card (VC), Universal Card Service (UCS), Mall'88 (ML88), Mall'91 (ML91) and Voice Recognition Call Processing (VRCP) (Buhrke et al., 1994; Chou et al., 1995).

Using a spoken dialogue system imposes a new set of challenges to the problem of recognizing digits embedded in natural spoken input. As opposed to using a system-initiative strategy, as it is the case for the databases reported under "READ" and "FLUENT", having a mixed-initiative dialogue implies less language constraints

[1] The digit error rate includes the insertion, deletion and substitution error rates normalized by the total digit count.

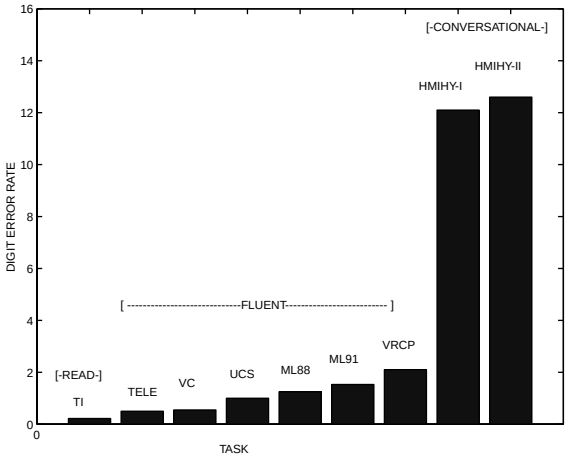

Fig. 1. Digit error rates for a variety of in-house data collections.

and more flexibility for users to speak openly (Chu-Carroll, 1999). Although this is clearly advantageous for building systems that facilitate natural human–machine communication, it can have negative impact on the performance of the recognizer. This is precisely the scenario that we are examining in this study in which during the course of the dialogue, users are asked a variety of questions among which "*What number would you like to call?*", "*May I have your card number please?*", "*What was that number again?*", etc. The difficulty here is *not only* to deal with fluent unconstrained speech, but to be able to design systems that can accurately recognize an *entire* string of numbers that may be encoded by digits, natural numbers and/or alphabets. In addition, these systems must be robust towards out-of-vocabulary words, hesitation, false-start and various other acoustic and language variabilities. Employing systems that accurately recognize merely digits in a spoken dialogue environment could lead to poor performances as illustrated in Fig. 1 ("CONVERSATIONAL") for the two "How May I Help You?" (HMIHY-I, HMIHY-II) data collections. [2]

[2] HMIHY-I and HMIHY-II are customer care recordings that were collected in two separate sessions.

3. Experimental study using a spoken dialogue system

We are interested in the problem of understanding fluent speech within constrained task domains. In particular, we have conducted a number of field trial studies on AT&T customers responding to the open-ended prompt "*How May I Help You?*" with the aim at providing an automated customer service. The goal of this service is to recognize and understand customers' requests whether they relate to billing, credit, call automation, etc. (Gorin et al., 1997; Riccardi and Gorin, 1999). Dialogue in this application is clearly necessary since in many situations involving ambiguous requests or poor system robustness, customer service can not be performed merely from a single input.

3.1. Database for numeric recognition

This paper will focus on specific parts of the dialogue where customers are prompted to say a credit card or a telephone number to obtain call automation or billing credit. Various types of prompts have been studied with the objective to stimulate maximally consistent and informative responses from large populations of users (Boyce and Gorin, 1996). These prompts are engineered towards asking users to say or repeat their number string without imposing rigid speaking format. Our experimental database included over 30,000 transactions with 2178 utterances representing responses to card and phone number queries, ranging from 1 to 45 words in length (referred to as the numeric database) (Riccardi and Gorin, 1999).

To calibrate the difficulty of this task, we subdivided the database based on two sets of results. The first, which is displayed in Fig. 2 in the form of pie charts, is a partitioning of the data according to three categories: (a) *numerics phrases only* (such as digits, natural numbers and alphabets) (b) *embedded numerics*, which include those numeric phrases that have been spoken among other vocabulary words, and (c) *no numerics*, which include utterances not containing any numeric phrases. The pie charts indicate that the distribution of users' responses based on spoken

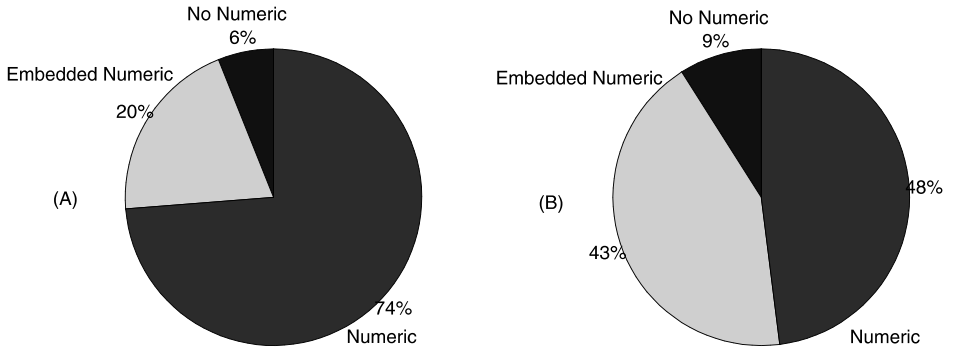

Fig. 2. Pie charts of users' responses to (A) card number, and (B) phone number prompts based on spoken numeric expressions.

numeric expressions is different for the card and phone number prompts. Furthermore, a large proportion of users do not respond with numeric phrases alone. In the case of the phone number prompt, 43% of the utterances contained embedded numeric and 9% included no numerics.

An alternative method for calibrating the difficulty of this task is illustrated in Fig. 3 which shows the classification of utterances as a function of their vocabulary contents and call characteristics (i.e, quality). Ten different classes are presented. They include *digits only* (''Digits'' such as ''one'', ''two'', ... , ''nine'', ''oh'', ''zero''); *embedded digits* (''eDigits'' – digits spoken among other vocabulary words); *natural numbers* (''nNumbers'' such as ''hundred'', ''eleven'' etc.); *alphabets* (''Alphas'' such as ''A'', ''H'', etc.); *restarts* (''Restart'' – false starts, hesitations and corrections); *accent* (''Accent'' – distinct regional dialect and strong foreign accent that contribute to mispronunciation of some words); *fast speech* (''Fast'' – over 1.5 times faster than the average speech rate); *noise* (''Noise'' – background speech, music and noise with signal-to-noise ratio below 15 dB); *out-of-vocabulary* (''Garbage'' – extraneous and uninformative speech); *cuts* (''Cuts'' – utterances with words that are partially spoken).

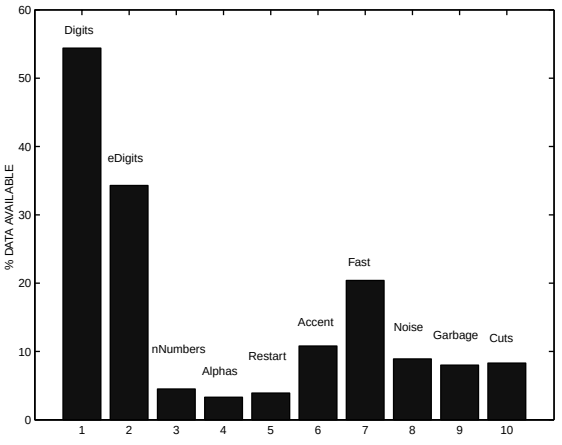

Fig. 3. Histogram of the numeric database as a function of vocabulary and call characteristics.

The statistics on this database are significantly different than most of the databases that we have previously encountered. Nearly half of the utterances include only digits as opposed to almost 100% for the databases reported by Chou et al. (1995). The new challenge this database presents is the need to accurately recognize embedded digits, natural numbers, alphabets, restarts and extraneous speech which collectively constitute about half of the database. There are also high proportions of fast speech, cuts and severe background noise which must all be dealt with appropriately.

3.2. The numeric language

It is not necessary for a spoken dialogue system to recognize all words correctly in order to understand the meaning of the request, but only those words or phrases that are ''salient'' to the task (Gorin, 1995; Wright et al., 1997, 1998). Salient phrases are essential for interpreting fluent speech. They may be automatically acquired such

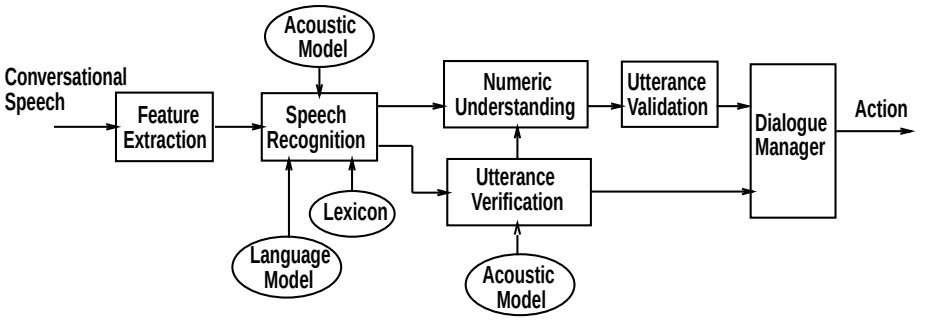

Fig. 4. A block diagram of the numeric recognition system.

that to maximize the accuracy of the mapping from unconstrained input to machine action (Wright et al., 1997).

In this paper, we will refer to those salient phrases that are relevant to our task as *numerics*. The *numeric language* consists of the set of phrases that are relevant to the task of understanding and interpreting customers' requests. For this application, we define six distinct phrase classes including *digits*, *natural numbers*, *alphabets* (Mitchell and Setlur, 1999), *restarts*, *city/country name*, and other *miscellaneous* phrases. Digits, natural numbers and alphabets are the basic building blocks of telephone and credit card numbers. For example, "*my card number is one three hundred fifty five A four...*". Restarts include the set of phrases that are indicative of false-starts, corrections and hesitation. For example, "*my telephone number is nine zero eight I'm sorry nine seven eight...*". City/country names can be essential in reconstructing a telephone number when country or area codes are missing. For example, "*I would like to call Italy and the number is three five...*". Finally, there are a number of miscellaneous salient phrases that can alter the ordering of the numbers. Such phrases are "area code", "extension number", "expiration date", etc. In our application, the vocabulary size was over 3600 words of which 100 phrases represented the numeric language.

4. System for numeric recognition

We consider the problem of numeric recognition in spoken dialogue systems as a large-vocabulary continuous speech recognition task where numerics are treated as a small subset of the active vocabulary in the lexicon. [3] The challenge here is essentially to accurately model, recognize and understand the numeric language in fluent unconstrained speech. Fig. 4 shows a block diagram of the numeric recognition system. The main components of this architecture are described as follows.

4.1. Feature extraction

The input signal, sampled at 8 kHz, is first pre-emphasized and grouped into frames of 30 ms durations at every interval of 10 ms. Each frame is Hamming windowed, Fourier transformed and then passed through a set of 22 triangular band-pass filters. Twelve mel cepstral coefficients are computed by applying the inverse discrete cosine transform on the log magnitude spectrum. To reduce channel variations while still maintaining real-time performance, each cepstral vector is normalized using cepstral mean subtraction with an operating look-ahead delay of 30 speech frames. To capture temporal information in the signal, each normalized cepstral vector along with its frame log energy are augmented with their first and second-order time derivatives. The energy coefficient, normalized at the operating look-ahead delay, is also applied for end-pointing the speech signal (Rabiner and Juang, 1993).

[3] Our previous experiments have shown that this approach to numeric recognition can lead to improved system performance over keyword detection methods (Rahim, 1999a).

4.2. Acoustic modeling

Accurate numeric recognition in fluent unconstrained speech clearly demands detailed acoustic modeling of the numeric language. It is also essential to accurately model non-numeric words as they constituted over 11% of the numeric database. Accordingly, our design strategy has been to use two sets of subword units; one dedicated for the numeric language and the other for the remaining vocabulary words. Each set adopts left-to-right continuous-density hidden Markov models (HMMs) with no skip states.

For numeric recognition, context-dependent acoustic units have been used which captured all possible inter-numeric coarticulation (Pieraccini and Rosenberg, 1990; Lee et al., 1992). The basic philosophy is that each word is modeled by three segments; a head, a body and a tail. A word would have one body, that has relatively stable acoustic characteristics, and multiple heads and tails depending on the preceding and following context. Thus junctures between numerics are explicitly modeled. Since this results in a huge number of subword units, and due to the limited amount of training data, the head-body-tail design has been strictly applied for the eleven digits. This generated 273 units which were assigned a 3-4-3 state topology corresponding to the heads, bodies and tails, respectively (Pieraccini and Rosenberg, 1990; Lee et al., 1992).

The second set of units were used for modeling non-numeric words and consisted of forty 3-state HMMs, each corresponding to a context-independent English phone (Shoup, 1980). Therefore, in contrast to traditional methods for digit recognition, out-of-vocabulary words – essentially the non-numeric words, are explicitly modeled by a dedicated set of subword units, rather than being treated as "filler" phrases. Transitional events between these units and the digit HMMs (limited by the availability of the data) were modeled by 16 additional units representing context switching from digits to speech events and vica versa. As will be shown later, this approach enables us to take full advantage of the language model for this task while maintaining real-time operation.

To model silence, background noise and extraneous events, four filler models were introduced. Each model was represented by either a 1-state or a 3-state HMM. They were trained on pre-segmented data that included silence, coughs, clicks, extraneous sounds and background speech.

In total, our system employed 333 units. Each state included 32 Gaussian components with the exception of the filler models which consisted of 64 Gaussian components. A unit duration model, approximated by a gamma distribution, was also used to increment the log likelihood scores (Rabiner and Juang, 1993).

4.3. Language modeling

Robustness is a key ingredient in building language models for spoken dialogue systems. The challenge is being able to construct models that can recognize fluent spontaneous speech, enabling users the flexibility to speak freely. Though the numeric language has quite a small vocabulary of very frequent words, the diverse and unpredictable set of responses make this task a challenge for language modeling. Thus traditional methods that rely on hand-crafted and deterministic grammars are generally inferior in these circumstances (Rahim, 1999a). On the other hand, training stochastic language models on transcriptions of responses to number requests is prone to data sparseness problems.

In this study, we automatically learn stochastic finite state machines from the training data using a variable n-gram model (Riccardi et al., 1996). This particular design includes back-off mechanism and enables parsing of any arbitrary sequence of words sampled from a given vocabulary. It has been previously shown to be more efficient than standard n-gram language models (Riccardi et al., 1996).

To improve the generalization capability of the language model, we have integrated semantic-syntactic knowledge into the estimate of the word sequence probabilities. In particular, by manually tagging the numeric language into word and phrase classes (e.g., $\langle digits \rangle = \{\text{ONE}, \text{TWO}, \ldots\}$, $\langle naturals \rangle = \{\text{ELEVEN}, \text{HUNDRED}, \ldots\}$, $\langle country \rangle = \{\text{ITALY}, \text{ENGLAND}, \ldots\}$), we are

effectively and robustly integrating prior knowledge into the language model. This has the benefit of producing probability estimates that are more robust against data sparseness, and language models that are both efficient in terms of storage and generalizable across different task domains (Riccardi et al., 1996).

Automatic learning of salient grammar fragments, such as classes of data, is essential for robust language modeling. Algorithms for automatic acquisition of phrase grammars through word and phrase classes have been proposed in (Riccardi and Bangalore, 1998). An excerpt of salient phrases that have been automatically generated from these algorithms are the following:

$\langle dig3\rangle$ *area code*,
number is $\langle dig10\rangle$,
$\langle country\rangle$ $\langle dig14\rangle$,

where $\langle dig3\rangle$, $\langle dig10\rangle$ and $\langle dig14\rangle$ are non-terminal symbols for different sequences of digits.

Incorporating language features as a constraint on the probability distribution estimation helps in improving the prediction of words over standard *n*-gram language models. For example the unigram model for the phrase $\langle dig3\rangle$*area code* is estimated from the following set of constraints on the word probability distribution:

$$\begin{aligned} &Pr_\theta(\langle dig3\rangle area\ code) = \theta_1\hat{p}_4 + \theta_2\hat{p}_1\hat{p}_2\hat{p}_3,\\ &Pr(area) = \hat{p}_1,\\ &Pr(code) = \hat{p}_2,\\ &Pr(\langle dig3\rangle) = \hat{p}_3,\\ &Pr(\langle dig3\rangle area\ code) = \hat{p}_4, \end{aligned} \tag{1}$$

where $\hat{p}_i$ are the priors and $\Theta = \{\theta_i\}$ are the free parameters which may be estimated through either the expectation-maximization algorithm or the cross-validation algorithm (Riccardi et al., 1996).

4.4. Model training

Training acoustic and language models for numeric recognition in a spoken language dialogue poses several new challenges. First, the training objective function should be coupled with the performance of the recognizer. For numeric recognition, an optimum recognizer is defined to be the one that minimizes the expected numeric string error rate. This performance measure is essential for speech understanding purposes since a numeric string would be considered erroneous if and only if any one of the numerics is recognized incorrectly. A misrecognition of a single digit, for example, would clearly result in an incorrect automation of a credit card or a telephone number.

The second challenge is to design a framework that provides "flexible" interaction between the acoustic and language models. Though one needs to maximize the recognition performance in a task-specific domain, flexibility must be provided to enable acoustic model training to be performed relatively freely of the constraints imposed by the stochastic language model.

Training is carried out in two phases using all the available training corpus, $\boldsymbol{X}$; Maximum likelihood estimation (MLE) is performed followed by minimum classification error (MCE) training (Juang and Katagiri, 1992). Given a set of language model parameters, Θ, the objective in MLE training is to learn a new set of acoustic model parameters, $\hat{\Lambda}$, through maximizing a log likelihood function. Given the correct word transcription, W_0, then

$$\hat{\Lambda} = \arg\max_{\Lambda} g(\boldsymbol{X}, W_0; \Lambda, \Theta), \tag{2}$$

where

$$g(\boldsymbol{X}, W_0; \Lambda, \Theta) = \log\left[\Pr(\boldsymbol{X}|W_0, \Lambda)\cdot \Pr(W_0|\Theta)^{\gamma}\right] \tag{3}$$

and $\gamma \geqslant 0$ defines a compensation factor which is set empirically to weight the contribution of the stochastic language model.

In MCE training, we aim to re-estimate both $\hat{\Lambda}$ and $\hat{\Theta}$ by minimizing a smoothed string-based error function:

$$(\hat{\Lambda}, \hat{\Theta}) = \arg\min_{\Lambda,\Theta}\{1 + \mathrm{e}^{-\alpha d(\boldsymbol{X};\Lambda,\Theta)}\}^{-1}, \quad \alpha > 0, \tag{4}$$

where $d(\boldsymbol{X}; \Lambda, \Theta)$, the misclassification distance, is defined as

$$d(\boldsymbol{X};\Lambda,\Theta) = -g(\boldsymbol{X},W_0;\Lambda,\Theta) + \log\left\{\frac{1}{N}\sum_{n=1}^{N} e^{g(\boldsymbol{X},W_n;\Lambda,\Theta)}\right\}. \quad (5)$$

W_n are considered as competing hypotheses and can be generated by an N-best search. [4]

Training and acoustic modeling are performed iteratively. Two sets of context-independent subword units are initially optimized using MLE followed by MCE. Context-dependent HMMs are then designed for the numeric language and trained accordingly. In the final step, additional units are integrated to model transitional events between numerics and non-numerics, and training is repeated again. The resultant acoustic model consisted of 333 discriminatively-trained HMMs.

Few remarks are in place for MCE training:

1. Discriminative training is relatively fast. With an efficient implementation of MCE along with fast N-best search, models have been trained at about real-time.
2. The objective function in Eq. (4) minimizes the expected string error rate over the training data. We have observed that assigning a dedicated set of context-dependent units for modeling numerics and another for modeling other words, as opposed to a single set of units for all vocabulary words, results in a lower error rate over both the training and test data.
3. The out-of-vocabulary words, which amount to 1% of our training database, were used for improving the discrimination between numerics and the filler models. This effectively reduces both the insertion and deletion rates of the numeric language.
4. The factor γ in Eq. (3) played a key role during discriminative training in providing flexible interaction between the acoustic and language models. The higher the value of γ, the more emphasis was given to the language model. Therefore, setting γ to be reasonably low during training provided the acoustic models the freedom to be optimized with less language constraints. [5]

4.5. Utterance verification

An important charter of a robust spoken dialogue system is identifying out-of-vocabulary utterances and utterances that are incorrectly recognized. This is particularly important for numeric recognition since it provides the dialogue manager with a confidence measure that may be used for call confirmation, repair or disambiguation. Associating each utterance with a confidence score is performed through utterance verification (Lleida and Rose, 1995; Rahim et al., 1997; Wendemuth et al., 1999). In this section, we will describe utterance verification as a statistical hypothesis testing problem. We will then describe our methods for extracting verification features and performing classification.

4.5.1. Statistical hypothesis testing

Consider $\boldsymbol{Z} = \{Z_1, Z_2, \ldots, Z_n\}$ to be n verification features that describe a single utterance having M words, $\{w_i | i = 1 : M\}$. In statistical hypothesis testing, if H_0 is the *null* hypothesis and H_1 is the *alternate* hypothesis, then the decision rule is stated in terms of a likelihood ratio as

$$\mathscr{L}(\boldsymbol{Z},\Theta) = \frac{Pr_{\theta_0}(\boldsymbol{Z}|H_0)}{Pr_{\theta_1}(\boldsymbol{Z}|H_1)} \underset{H_1}{\overset{H_0}{\gtrless}} \tau, \quad (6)$$

where τ is the decision threshold and $\Theta = \{\theta_0, \theta_1\}$ are the parameters of the model which are estimated by minimizing the average *probability of error* $\Pr_{\theta_1}(\boldsymbol{Z}, H_0) + \Pr_{\theta_0}(\boldsymbol{Z}, H_1)$ (Bickel and Doksum, 1977). If $\mathscr{S}^n$ denotes the numeric language set, then H_0 assumes that $w_i \in \mathscr{S}^n$, for at least one word, and that all numeric words are correctly recognized. Conversely, H_1 assumes that either $w_i \notin \mathscr{S}^n$ $\forall i$, or at least one of the numeric words is incorrectly recognized.

In this study, we consider utterance verification as a binary classification problem where for each

[4] In this study, MCE has been limited to training the acoustic model parameters only.

[5] In this study, γ was set to 7 during training and 9 during recognition thus causing the N-best search during the training process to be more dominated by the acoustic likelihood score.

utterance, the most probable hypothesis is selected as

$$H^* = \arg\max_{i=0,1} \Pr_{\phi}(H_i|\boldsymbol{Z}). \tag{7}$$

4.5.2. Verification features

Verification features, $\boldsymbol{Z}$, are designed to reflect the confidence in recognizing essentially the numeric language. Five features are used to encode each utterance. Pilot experiments have shown that each of these features has the capability of providing some degree of separation between H_0 and H_1. This is evident since the equal error rate is significantly better than chance level when employing any of the verification features individually (Rahim et al., 1997).

Verification score. To detect out-of-vocabulary words and numerics that are incorrectly recognized, a log likelihood ratio distance is computed based on a set of discriminatively-trained context-independent verification HMMs, Ψ, involving numeric phrases, $\Psi^{(k)}$, anti-numeric phrases, $\Psi^{(ak)}$, and fillers, $\Psi^{(f)}$ (Rahim et al., 1997). Let $\boldsymbol{X}$ be the acoustic features for a given utterance (e.g., cepstrum, energy, etc.), $\{w_k|k=1:K\}$ its numeric words, such that $w_k \in \mathscr{S}^n \ \forall k$, and $\boldsymbol{X}_k$ the corresponding acoustic vectors, then the verification score is defined as

$$L(\boldsymbol{X},\Psi) = \frac{1}{\eta}\log\left\{\frac{1}{K}\sum_k \exp\{-\eta L(\boldsymbol{X}_k,\Psi)\}\right\},$$

$$L(\boldsymbol{X}_k,\Psi) = \log\left\{\frac{\Pr(\boldsymbol{X}_k|\Psi^{(k)})}{\alpha\Pr(\boldsymbol{X}_k|\Psi^{(ak)}) + (1-\alpha)\Pr(\boldsymbol{X}_k|\Psi^{(f)})}\right\}, \tag{8}$$

where $\eta > 0.0$ and $0 \leqslant \alpha \leqslant 1$. This measure has been commonly used in (Lleida and Rose, 1995; Rahim et al., 1997).

N-best verification score. This score reflects the confidence in $L(\boldsymbol{X},\Psi)$, estimated over the N-best candidates:

$$dL(\boldsymbol{X},\Psi) = L_1(\boldsymbol{X},\Psi) - L_2(\boldsymbol{X},\Psi), \tag{9}$$

where $L_1(\cdot)$ and $L_2(\cdot)$ are the verification scores for the best two candidates.

Likelihood-ratio distance. The ratio of the N-best likelihood scores for the recognition HMMs, Λ, provides a confidence measure that can describe the best decoded path.

$$dL(\boldsymbol{X},\Lambda) = L_1(\boldsymbol{X},\Lambda) - L_2(\boldsymbol{X},\Lambda),$$
$$L(\boldsymbol{X},\Lambda) = \frac{1}{K}\sum_k \log\{\Pr(\boldsymbol{X}_k, w_k|\Lambda)\}, \tag{10}$$

where $L_1(\cdot)$ and $L_2(\cdot)$ are the likelihood scores for the two best candidates.

Numeric cost function 1. Confusions among the numeric language for the top N-best candidates is a strong indication of a possible error. Thus for $w_k \in \mathscr{S}^n$,

$$D_1 = \begin{cases} 1 & w_k^1 = w_k^2 \quad \forall k, \\ 0 & \text{otherwise}, \end{cases}$$

where w_k^1 and w_k^2 are the numeric words for the two best candidates.

Numeric cost function 2. Since for some utterances the numeric language can be the only vocabulary in the N-best candidates, D_1 will erroneously be set to 0 in these situations. A companion cost is therefore introduced, such that

$$D_2 = \begin{cases} 1 & w_k^1 \in \mathscr{S}^n \quad \forall k, \\ 0 & \text{otherwise}. \end{cases}$$

4.5.3. Hierarchical mixture-of-experts

The action of the classifier is to map the input feature space into H_0 should the top hypothesis contain numeric words that are *all* correctly recognized, and H_1 otherwise. A hierarchical mixture-of-experts (HME) has been used for this binary classification problem (Jordan and Jacobs, 1994). Unlike neural networks, HMEs apply the expectation-maximization algorithm for training as opposed to gradient descent. HMEs are also *discriminative*, thus unlike Gaussian mixture models (GMM) that maximize a likelihood function, HMEs minimize the probability of error. They generally converge reasonably fast accepting both binary and continuous inputs.

Fig. 5 shows an HME of depth 2. This architecture enables non-linear supervised learning by dividing the input feature space into a nested set of regions and fitting simple surfaces to the data that

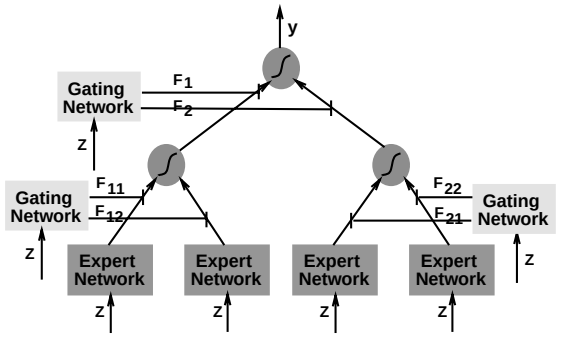

Fig. 5. A hierarchical mixture-of-experts (after Jordan and Jacobs, 1994).

fall in these regions. The output of the HME, y, is a discrete random variable having possible outcomes of 1, to denote H_0, or 0 to denote H_1.

An HME is a binary tree having *gating networks* that sit at the nonterminals of the tree and *expert networks* that sit at the leaves of the tree. Both gating and expert networks receive the verification features $\boldsymbol{Z}$ and compute a conditional probability. For binary classification, the resulting hierarchical probability model is a mixture of Bernoulli densities (Jordan and Jacobs, 1994).

$$\Pr_{\phi}(y|\boldsymbol{Z}) = \sum_i F_i(\boldsymbol{Z},\phi_i)\sum_j F_{ij}(\boldsymbol{Z},\phi_{ij}) \times \Pr_{\phi_{ij}}(y|\boldsymbol{Z})^{y}(1-\Pr_{\phi_{ij}}(y|\boldsymbol{Z}))^{(1-y)}. \quad (11)$$

Thus, the probability of selecting H_0 is

$$\Pr_{\phi}(y=1|\boldsymbol{Z}) = \sum_{i=0,1} F_i(\boldsymbol{Z},\phi_i)\sum_j F_{ij}(\boldsymbol{Z},\phi_{ij})\Pr_{\phi_{ij}}(y|\boldsymbol{Z}), \quad (12)$$

where ϕ are the underlying parameters of the HME. The output of the HME may be considered as a confidence score. Based on the application design requirements, this output can be adjusted so that to achieve a desirable trade-off between false acceptance and false rejection rates. [6]

[6] From a business point of view, threshold settings are directly related to call automation rate and hence cost revenue. The higher the threshold, the smaller is the automation rate but the less are the errors (false acceptance) made by the system, and vica versa.

4.6. Numeric understanding

Speech, or language, understanding is an essential component in the design of spoken dialogue systems. It provides a link between the speech recognizer and the dialogue manager (see Fig. 4) with the prime responsibility of converting the recognition output into a machine action. [7]

For numeric recognition, the purpose of the understanding module is to translate the output of the recognizer into a "valid" string of digits. However, in the event of an ambiguous request or poor recognition performance, the understanding module may provide several hypotheses to the dialogue manager for *repair*, *disambiguation* or *clarification* (Abella and Gorin, 1997).

In this study, a knowledge-based strategy has been implemented for numeric understanding to translate the recognition results (e.g., N-best hypotheses) into a simplified finite state machine (FSM) containing digits only. Six semantic classes for numeric understanding are illustrated in Table 1. A simplified example is shown in Fig. 6 which illustrates the basic actions of the understanding module when dealing with alphabets, natural numbers, restarts or corrections, and filtering of out-of-vocabulary phrases. It should be pointed out that the knowledge-based system is very much suited for our task given its scope and complexity. However, if sufficiently more data had been available, it would have been interesting to explore machine learning methods for this problem.

One variation of the understanding module is to process N-best hypotheses (or a lattice). This provides the dialogue system with much flexibility as to what the appropriate action to take especially if the top hypothesis is not a valid string as will be illustrated in the next section.

4.6.1. String validation

Speech recognition systems employ language modeling for encoding knowledge of the task. This is advantageous from both accuracy and efficiency standpoints. In this study class based language

[7] For our particular task domain, we will refer to speech understanding as numeric understanding.

Table 1
Semantic classes for numeric understanding

Class	Action	Example
Digits	Translation	*Six seven eight → 6 7 8*
Naturals	Translation	*One eight hundred two one three → 1 8 0 0 2 1 3*
Restarts	Correction	*Nine zero eight sorry nine one eight → 9 1 8*
Alphabets	Translation	*A Z one two three → 2 9 1 2 3*
City/country	Translation	*Calling Florham Park New Jersey → 9 7 3*
Numeric phrases	Realignment	*Nine one two area code nine zero one →9 0 1 9 1 2*
Out-of-vocabulary	Filtering	*I do not know what you are talking about → ϕ*

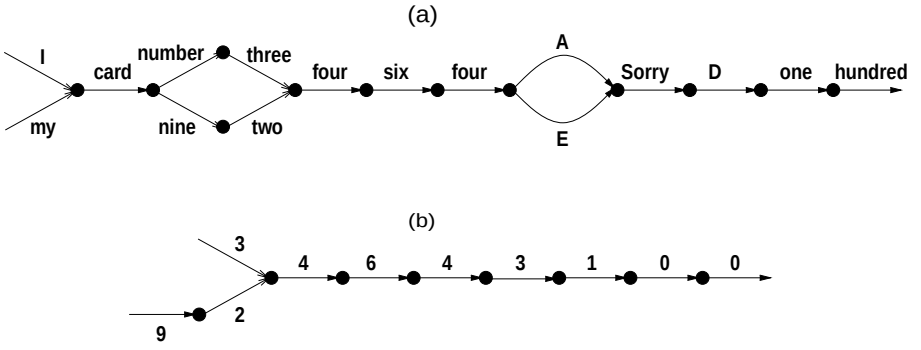

Fig. 6. An example of an FSM: (a) before and (b) after numeric understanding.

models were used to integrate semantic-syntactic knowledge into the estimate of the word sequence probabilities. However, no constraints within the numeric classes were imposed, that is, users had the flexibility to speak the numeric language in any ordering manner. For example users can say a card number intermixing alphabets, digits and natural numbers. In these situations the system would rely more strongly on the acoustic scores during decoding.

The motivation for this design strategy was that users who commonly have no previous experience of the system would unlikely follow any language structure. This is clearly a challenge when dealing with a mixed-initiative dialogue, as opposed to a system-initiative dialogue. We observed many instances where the user decided either to change context, not to provide a complete telephone or credit card number, or wished not to provide any information and requested an operator. These varieties of problems are real and very much expected from a mixed-initiative dialogue system that deals with novice users. In the test database that was used in this study, we found nearly 24% of the calls to fit into this category, prohibiting full automation of the call accordingly.

It is essential that these problematic situations are identified and reported to the dialogue manager. In this study, the output of the understanding system is passed through a validation module. This module performs a database query which accesses live databases of credit cards and telephone numbers, reporting three possible outcomes:

1. *Valid.* The output digit string of the understanding system corresponds to an actual valid number (telephone or credit).
2. *Invalid.* The output digit string does not correspond to a valid number.
3. *Partially valid.* The output digit string is invalid but at most one digit may be corrected to ensure validity of the string.

This information is then passed to the dialogue manager for either confirmation, disambiguation or clarification.

Since a database of numbers can be represented by an FSM, it can be efficiently combined with the understanding FSM in Section 4.6. In this manner, N-best hypotheses may also be processed and passed to the dialogue manager. Hence, understanding and validation can be conducted simultaneously.

Performing validation on the output of the understanding module is essential in spoken language dialogue. There are many situations where this can be advantageous. A speaker may provide an invalid area code, for example, or a valid telephone number that was either incorrectly recognized or misinterpreted by the system. Clearly doing a checking step prior to automation is necessary.

Finally, it should be pointed out that this process of validation after recognition is different than traditional approaches that constrain the recognition process by means of a number grammar. This latter approach is unsuitable in spoken language dialogue where errors can be made by either the speaker or the machine.

5. Experimental results

This section provides experimental results demonstrating the performance of the various modules of our system, including recognition, understanding, verification and validation. All experiments have been performed using AT&T Watson speech recognition system (Sharp et al., 1997). A standard Viterbi beam search has been used with a lexicon of 3.6 K words, perplexity 14 and out-of-vocabulary word rate of less than 5% as measured on the test database (Riccardi and Gorin, 1999).

A variety of databases have been used for training the subword/numeric models. These databases, collected over a wide range of environmental conditions, help to provide broad acoustic models. They include (a) 12,000 utterances from this experimental study (of the total 30,000 utterances) with over 1500 utterances from users responding to a card or a telephone number prompt (Gorin et al., 1997), (b) 11,500 connected digits strings from services and data trials performed over a 10 year period (Chou et al., 1995), (c) 3300 connected digits strings, from various cellular environments (acquired from BRITE systems), (d) an in-house database including 5000 strings of connected digits and natural numbers.

For language modeling, a variable n-gram stochastic automata was estimated using the 30,000 transactions. Two separate language models were then generated for the card and phone numbers by adapting on their respective transcriptions (Riccardi and Gorin, 1999).

Our experiments have been conducted on 626 test utterances, each representing a separate transaction from a different speaker. The first set of results are presented in Fig. 7. They illustrate the performance of the recognizer for two different acoustic models. The first model, represented by the dashed line, consists of two sets of MLE-trained context-independent HMMs. This model was used in the initial stage of the training process as pointed out in section 4. The solid line represents the MCE-trained context-dependent model. The figure presents the average overall word error rate and the digit error rate (with no rejection) as a function of processing speed on an SGI R10000 machine. [8] Varying the speed of the decoder has been obtained by changing the operating beam width.

From the two results in Fig. 7, the following conclusions can be drawn. (a) Improved acoustic modeling and discriminative training has led to 20–30% reduction in the word and digit error rates. The reduction in the digit error rate, in particular, is critical since it translates to over 10% reduction in the absolute numeric string error rate (Rahim, 1999a). (b) For either method, the knee points on the curves are well positioned below the real-time (RT) mark. The computing time shown in Fig. 7 represents off-line processing of the entire system in Fig. 4.

The second set of experiments evaluates the speech understanding module. At this stage, we

[8] The word error rate includes the insertion, deletion and substitution error rates normalized by the total word count. The digit error rate includes the insertion, deletion and substitution error rates at the output of the understanding module normalized by the total digit count.

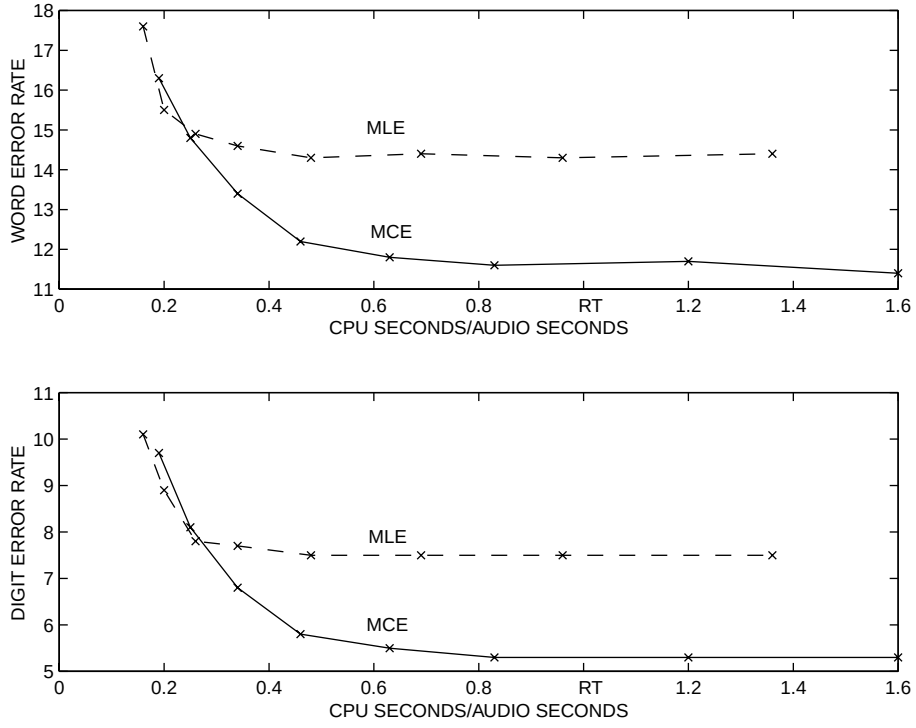

Fig. 7. Word and digit error rates versus processing speed per audio second. The dashed and solid lines represent the performance of the MLE context-independent and the MCE context-dependent acoustic models, respectively.

are interested whether the interpretation of the spoken input following recognition matches that of the transcription. The understanding module converts an input text into a sequence of digits using the rules defined in Table 1. An action of this module is considered correct if the output digit sequences match between the recognized speech and its transcription. Fig. 8 displays the understanding error rates for the two previously described models (i.e., MLE-based and MCE-based) as a function of processing speed. These results echo our previous findings that a discriminatively trained context-dependent acoustic model outperforms a standard ML-trained context-independent model. A typical operating point is 26% error rate with no rejection. This implies that the interpretation for 74% of the utterances is the same for the recognized speech and its transcription.

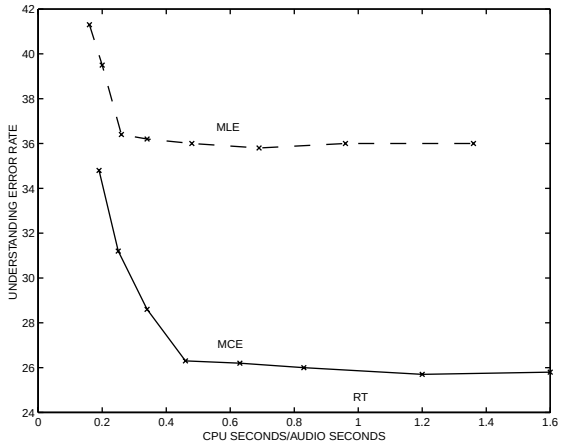

Fig. 8. Understanding error rate versus processing speed per audio second. The dashed and solid lines represent the performance of the MLE context-independent and the MCE context-dependent acoustic models, respectively.

An important strategy for improving system performance is by utilizing utterance verification. In the third set of experiments we aim to utilize the output of the HME as a confidence score to identify when the interpretation of the understanding module is incorrect. Clearly this provides

valuable information for the dialogue manager when responding to the user. In this study, we have applied utterance verification for detecting recognition errors among utterances that included the numeric language. This is clearly a much tougher challenge than rejecting out-of-vocabulary words. Utterances that included solely out-of-vocabulary words were found not to be confused with the numeric language strictly based on their recognized contents. Separating and rejecting those utterances was therefore trivial without having to apply utterance verification. This conclusion is considered as one of the advantages of the proposed approach for numeric recognition. Methods that rely on more constrained grammars often show larger confusability between in-vocabulary and out-of-vocabulary data.

Fig. 9(b) shows the distribution of the output of the HME when the data is correctly interpreted, i.e., $\Pr_\phi(y = H_0|\boldsymbol{O})$, and when the data is incorrectly interpreted, i.e., $\Pr_\phi(y = H_1|\boldsymbol{O})$. To baseline our results, Fig. 9(a) shows the distribution of the likelihood ratio scores when computed using the verification models. This is considered as the classical approach when performing utterance verification, particularly for digit recognition (Rahim et al., 1997). The amount of overlap in the two distributions, which define the false rejection rate and the false acceptance rate, is clearly smaller for the HME case; a result that is attributed to the additional input features. In particular, the equal error rate (i.e., false acceptance = false rejection) and the minimum error rate (min(false acceptance + false rejection)) are 14% and 23%, respectively, for the likelihood ratio score, and 10% and 18%, respectively, for the HME method.

In the following results we compare the performance of the HME with that of a GMM. Fig. 10 shows the receiver operating characteristic (ROC) curves for the two methods, that is, the

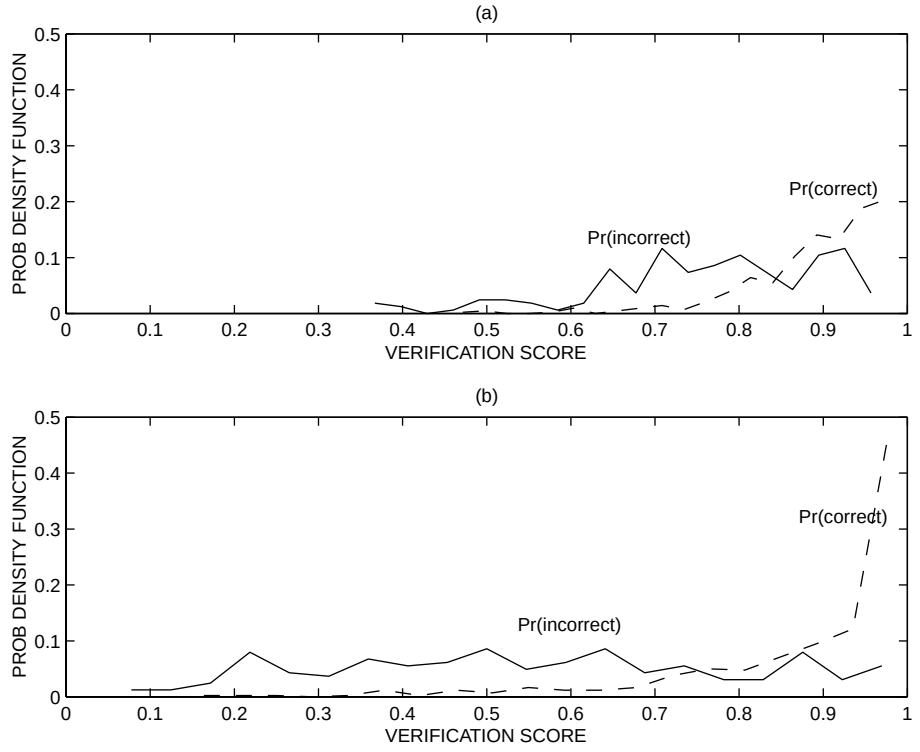

Fig. 9. Utterance verification performance when using (a) a likelihood ratio score, and (b) HME verification score. The solid and dashed lines represent the distribution of the scores when the data is incorrectly and correctly interpreted, respectively.

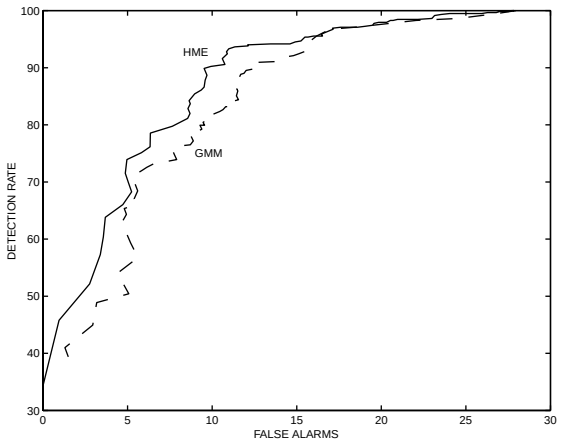

Fig. 10. ROC curves when using the GMM and the HME for utterance verification.

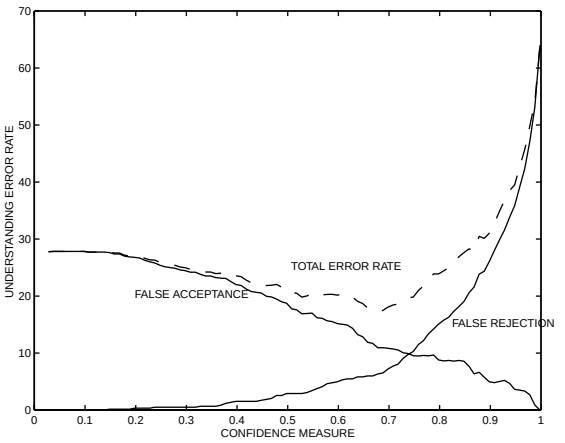

Fig. 11. False acceptance, false rejection and total error rates as a function of the decision threshold (confidence measure).

detection rate (1– false rejection rate) versus false alarms (false acceptance). The HME clearly demonstrates better performance than the GMM at all possible settings. At an operating point of 10% false acceptance rate, for example, the detection rates are 79% and 90% for the GMM and the HME systems, respectively. Instrumenting utterance verification at this operating point translates to a boost in the understanding correct rate for the HME system from 74%, as shown in Fig. 8 when not using utterance verification, to 90% (Rahim, 1999b).

From a designer's prospective, Fig. 11 illustrates the trade-off between false acceptance and false rejection as a function of the confidence measure (or decision threshold). One interesting remark is that the output of the HME is a true probability which reflects a mapping of the verification measurements into either correct or incorrect class decision. The minimum and equal error rates, which both seem to have similar thresholds, are less than 18% and 10%, respectively. The last step before sending information to the dialogue manager is validation. Checking whether the sequence of digits at the output of the understanding system corresponds to an active telephone or card number is valuable information. Based on this knowledge, the dialogue manager may determine a particular strategy, such as to reprompt the user or to simply connect to a live operator.

Fig. 12 shows a break down of the telephone number strings based on whether or not they represented valid or invalid numbers. Validity of these strings was a result of an inquiry with a telephone number database. These telephone numbers were acquired at the output of the understanding system. The figure shows the results for the top 10 candidate strings which were checked against the database in order. Should the

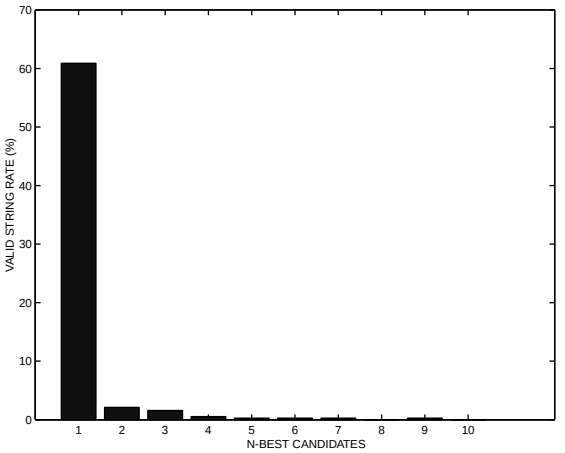

Fig. 12. Percentage of valid telephone number strings over the top 10 best candidates.

Table 2
Digit and string error rates for valid and invalid telephone numbers

	Valid (top candidate)	Invalid
Digit	0.6	18.5
String	5.5	55.0

top candidate correspond to an invalid number, for example, the next candidate would be checked, and so. The results in Fig. 12 indicate that 61% of the recognized output strings (i.e., top hypothesis) correspond to valid telephone numbers. For the 39% invalid strings, the figure shows that succeeding candidates are valid in some instances. Based on this finding, the dialogue manager may decide to examine those candidates instead. [9]

Table 2 presents the digit and string error rates based on the top candidate for the valid and invalid data. For the 61% valid strings, the digit and string error rates are 0.6% and 5.5%, respectively. For the invalid strings, these error rates are 18.5% and 55.0%, respectively. Those remarkable results imply that valid telephone numbers at the output of the understanding system are very likely to be correctly recognized; a conclusion that is valuable in the design of spoken dialogue systems.

6. Discussion and summary

The problem we are trying to solve is automatic recognition and understanding of fluent speech in spoken dialogue systems. In particular, we are focusing on task domains that involve the use of the numeric language. For example, tasks that require the utilization of credit card, telephone numbers, zip codes, dates, times, etc. Irrespective of the application, it is our premise that these tasks are very well defined and that they can be integrated into a unified language framework.

The use of the numeric language in mixed-initiative spoken language dialogue presents several challenges particularly when dealing with users who are unfamiliar with the operation or the design of the dialogue system. First, the spoken number string may be encoded by digits, natural numbers and alphabets that must *all* be correctly recognized to successfully automate the call. Second, the input speech may be embedded in a pool of out-of-vocabulary words with possible corrections, false-starts and other problematic situations that are features of fluent and spontaneous speech. Third, the input speech may be ambiguous, that is, it may contain either conflicting information, invalid numbers or incomplete information. Rather than trying to deal with these problems in a general way, our approach has been to narrow down this problem to consider only those "salient" phrases that are of interest to our task. The set of these phrases were referred to as the numeric language.

This paper presented our progress towards robust numeric language recognition in spoken language dialogue. Our strategy has been to consider this problem as a large vocabulary speech recognition task that is especially tailored to provide high quality robust recognition of the numeric language. This approach is unlike traditional methods that instrument handcrafted grammars for digit recognition along with filler models to accommodate for out-of-vocabulary words.

The experimental database presented in this study represented automated customer service inquiries for credit card and telephone numbers. This is a subset of a field trial involving AT&T customers responding to the open-ended prompt "How May I Help You?". For this spoken language dialog, maintaining system robustness implies enabling *any* customer to access information when calling from *anywhere*. This target demands maintaining robustness at several levels of system design, including acoustic, language, understanding and dialogue. In this paper, we presented methods for acoustic modeling, discriminative training, class-based language modeling, utterance

[9] Performing the validation process on the transcribed data, as opposed to the recognized strings, resulted in 76% validity. This indicates that misrecognition in our system caused 15% (76 − 61) of the requests to be misinterpreted.

verification, numeric understanding and string validation. Collectively, these building blocks generate several string hypotheses which are passed to the dialogue manager along with confidence measures and information on whether or not these strings represent valid numbers. The action set of the dialogue manager includes call completion, disambiguation, confirmation, clarification and error recovery (Abella and Gorin, 1997).

Experimental results on 626 utterances have concluded the following:

1. Improved acoustic modeling and discriminative training for the numeric language leads to 20–30% reduction in the numeric error rate. It also provides a higher interpretation accuracy of the numeric data and reduces the absolute understanding error rate by 10%.
2. The combination of five verification features using an HME provides an accurate utterance verification system for identifying incorrectly interpreted strings. At an operating point of 10% false acceptance, for example, the detection rate increases from a baseline of 74% to over 90%.
3. Validating the output of the understanding system through a predefined grammar of telephone and card numbers helps tremendously in reducing the overall system error rate. For the 61% valid telephone numbers at the output of recognizer, the digit error rate was observed to be 0.6%.

It should be pointed out that the majority of the data used in our experiments were collected from field trial experiments with real customers. Accordingly the set of problems that the system encounters are more challenging than those experienced from data recordings with users familiar with either the technology or the system. Although the amount of test data used in this study was rather limited, it pointed out to some interesting problems that need to be addressed in spoken language dialogue. It also suggests that maintaining robustness in dialogue systems require tight integration and communication between the speech recognition module, the understanding module and the dialogue manager. Further research in this direction is clearly needed.

Acknowledgements

The authors would like to thank E. Bocchieri, R. Rose and the Watson development team for their technical help.

References

Abella, A., Gorin, A.L., 1997. Generating semantically consistent inputs to a dialog manager. In: Proceedings of the European Conference on Speech Communication and Technology, pp. 1879–1882.

Bickel, P., Doksum, K., 1977. Mathematical statistics: Basic ideas and selected topics. Prentice-Hall, Englewood Cliffs, NJ.

Boyce, S., Gorin, A.L., 1996, User interface issues for natural spoken dialogue systems. In: Proceedings of the International Symposium on Spoken Dialogue (ISSD), pp. 65–68.

Buhrke, E., Cardin, R., Normandin, Y., Rahim, M., Wilpon, J., 1994. Application of vector quantized hidden Markov models to the recognition of connected digit strings in the telephone network. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing.

Cardin, R., Normandin, Y., Millien, E., 1993. Inter-word coarticulation modeling and MMIE training for improved connected digit recognition. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pp. 243–246.

Chou, W., Rahim, M.G., Buhrke, E., 1995. Signal conditioned minimum error rate training. In: Proceedings of the European Conference on Speech Communication and Technology, pp. 495–498.

Chu-Carroll, J., 1999. Form-based reasoning for mixed-initiative dialogue management in information-query systems. In: Proceedings of the European Conference on Speech Communication and Technology, pp. 1519–1522.

Gorin, A.L., 1995. On automated language acquisition. J. Acoust. Soc. Am. 97 (6), 3441–3461.

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

Jordan, M., Jacobs, R., 1994. Hierarchical mixtures of experts and the EM algorithm. Neural Comput. 6, 181–214.

Juang, B.-H., Katagiri, S., 1992. Discriminative learning for minimum error classification. IEEE Trans. Acoust., Speech, Signal Process. 40, 3043–3054.

Lamel, L., Rosset, S., Gauvain, J.-L., Bennacef, S., 1999. The LIMSI ARISE system for train travel information. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing.

Lee, C.-H., Giachin, E., Rabiner, L.R., Pieraccini, R., Rosenberg, A.E., 1992. Improved acoustic modeling for large vocabulary continuous speech recognition. Computer Speech Language 6 (2), 103–127.

Lleida, E., Rose, R.C., 1995. Utterance verification in continuous speech recognition. In: Proc. IEEE ASR Workshop.

Mitchell, C., Setlur, A., 1999. Improved spelling recognition using a tree-based fast lexical match. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing.

Os, E., Boves, L., Lamel, L., Baggia, P., 1999. Overview of the ARISE project. In: Proceedings of the European Conference on Speech Communication and Technology, pp. 1527–1530.

Pieraccini, R., Rosenberg, A.E., 1990, Coarticulation models for continuous digit recognition. In: Proceedings of the Acoustics Society of America, p. 106.

Rabiner, L., Juang, B.-H., 1993. Fundamentals of speech recognition. Prentice-Hall, Englewood Cliffs, NJ.

Rahim, M., 1999a. Recognizing connected digits in a natural spoken dialog. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing.

Rahim, M., 1999b. Utterance verification for the numeric language in a natural spoken dialogue. In: Proceedings of the European Conference on Speech Communication and Technology, pp. 495–498.

Rahim, M., Juang, B.-H., 1996. Signal bias removal by maximum likelihood estimation for robust speech recognition. IEEE Trans. Speech Audio Process. 4 (1), 19–30.

Rahim, M., Juang, B.-H., Chou, W., Buhrke, E., 1996. Signal conditioning techniques for robust speech recognition. IEEE Signal Process. Lett. 3 (2), 107–109.

Rahim, M., Lee, C.-H., Juang, B.-H., 1997. Discriminative utterance verification for connected digits recognition. IEEE Trans. Speech Audio Process. 5 (3), 266–277.

Rahim, M., Pieraccini, R., Eckert, W., Levin, E., Di Fabbrizio, G., Riccardi, G., Lin, C., Kamm, C., 1999. W99 – a spoken dialogue system for the ASRU99 workshop. In: Proc. IEEE ASRU Workshop.

Ramaswamy, G., Kleindienst, J., Coffman, D., Gopalakrishnan, P., Neti, C., 1999. A pervasive conversational interface for information interaction. In: Proceedings of the European Conference on Speech Communication and Technology, pp. 2662–2666.

Riccardi, G., Bangalore, S., 1998. Automatic acquisition of phrase grammars for stochastic language modeling. In: ACL Workshop on Very Large Corpora Proceedings, pp. 188–196.

Riccardi, G., Gorin, A.L., 1999. stochastic language adaptation over time and state in natural spoken dialog systems. IEEE Trans. Speech, Audio Process. 8, 3–10.

Riccardi, G., Pieraccini, R., Bocchieri, E., 1996. Stochastic automata for language modeling. Computer Speech and Language 10, 265–293.

Sankar, A., Lee, C.-H., 1996. A maximum-likelihood approach to stochastic matching for robust speech recognition. IEEE Trans. Speech Audio Process. 4 (3), 190–202.

Sharp, R.D., Bocchieri, E., Castillo, C., Parthasarathy, S., Rath, C., Riley, M., Rowland, J., 1997. The Watson speech recognition engine. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pp. 4065–4068.

Shoup, J., 1980. Phonological aspects of speech recognition. In: Lea, W. (Ed.), Trends in Speech Recognition. Prentice-Hall, Englewood Cliffs, NJ.

Wendemuth, A., Rose, G., Dolfing, J., 1999. Advances in confidence measures for large vocabulary. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing.

Wright, J.H., Gorin, A.L., Abella, A., 1998. Spoken language understanding within dialogs using a graphical model of task structure. In: Proc. ICSLP.

Wright, J.H., Gorin, A.L., Riccardi, G., 1997. Automatic acquisition of salient grammar fragments for call-type classification. In Proceedings on Eurospeech, pp. 1419–1422.