Semantic Confidence Measurement for Spoken Dialog Systems

Ruhi Sarikaya, *Member, IEEE*, Yuqing Gao, Michael Picheny, *Fellow, IEEE*, and Hakan Erdogan, *Member, IEEE*

***Abstract*—This paper proposes two methods to incorporate semantic information into word and concept level confidence measurement. The first method uses tag and extension probabilities obtained from a statistical classer and parser. The second method uses a maximum entropy based semantic structured language model to assign probabilities to each word. Incorporation of semantic features into a lattice posterior probability based confidence measure provides significant improvements compared to posterior probability when used together in an air travel reservation task. At 5% False Alarm (FA) rate relative improvements of 28% and 61% in Correct Acceptance (CA) rate are achieved for word level and concept level confidence measurements, respectively.**

I. Introduction

AUTOMATIC speech recognition systems are far from perfect. There are a number of factors including environment, telephone line quality, and speaker variability that can impair speech recognition performance. Moreover, in some cases a speech understanding component can generate an incorrect parse result sending the dialog on a completely wrong path. In order to circumvent these problems it is vital to employ a reliable confidence metric that can identify speech recognition errors. This information can be used to generate repair dialogs. Attaching a confidence value to each word can also be used for improved unsupervised adaptation, automatic weighting of speech and nonspeech information sources, and information retrieval algorithms. Furthermore, speech recognition confidence can be used in conjunction with the natural language understanding component of the system to affect the parsing strategy [16].

In [21] a number of issues regarding the incorporation of a confidence metric into a speech recognition system are listed. These issues include 1) at what stage the confidence metric should be applied, 2) the definition of what constitutes an error, 3) what are the most useful features to incorporate, 4) what model should be used to combine various features, and 5) how to measure the performance of a confidence metric. A significant portion of the research for confidence annotation methods for limited domains centers around items 3 and 4. The majority of the approaches share two basic steps: 1) generate as many features as possible based on the speech recognition and/or natural language understanding process and 2) use a classifier to combine these features in a reasonable way. Typically, confidence measures depend on the type of the task and the particular application. For domain independent large vocabulary speech recognition systems, posterior probability based on a word graph is shown to be the single most useful confidence feature [4], [15]. For limited domains, features from a speech understanding unit are also helpful.

There are a number of cues that suggest when a speech recognition hypothesis may be in error. These cues can be observed from acoustic model scores, language model scores, word counts in an N-best list, lattice density, phone perplexity, language model back-off behavior, and posterior probabilities [2], [12], [8], [14]. However, many of these features overlap considerably and they have already been included in the recognition process directly or indirectly. As a result, the combination of a number of features from the same source may only result in a marginal improvement over the best single feature.

In many, if not all, of the previous studies the way in which the semantic information is incorporated into the decision process is rather *ad hoc*. For example in [14], the semantic weights assigned to words were based on heuristics. Similarly, in [3] such semantic features as "uncovered word percentage," "gap number," "slot number," etc. were generated experimentally in an effort to incorporate semantic information into the confidence metric. Using the information derived from the concept sequence generated by the parser combined with the lattice posterior probability improved the confidence measurement performance. The improvement was modest on some tasks [8], [3] while more pronounced for the others [12]. Similar to using many overlapping speech recognition based features, here many overlapping semantic features are designed. Recently, there have been new approaches attempting to integrate lexical and semantic content of the sentence tightly. In [5], a word graph is converted into a concept graph and confidence measure is computed on the concept graph providing significant improvement over posterior probability based feature. Decoupling of language and acoustic modeling in speech recognition along with introduction of latent semantic analysis (LSA) to find (dis)similar words in a sentence for confidence measurement is also shown to provide improvement over traditional speech recognition based features [18]. However, we believe that significant research effort has to focus on developing a framework which allows for tight integration of the lexical and semantic content of a sentence.

Confidence measurement can be applied either at the word level, phrase/concept level, utterance level or their combinations. In this paper, both word, and concept level confidence annotations are considered. We propose two methods that use two sets of statistical features to model semantic information in a sentence. The first relies solely on the semantic classer/parse

Manuscript received April 18, 2003; revised March 10, 2004. The Associate Editor coordinating the review of this manuscript and approving it for publication was Dr. Hermann Ney.

The authors are with the Human Language Technologies Group, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: sarikaya@us.ibm.com; yuqing@us.ibm.com; picheny@us.ibm.com; haerdogan@sabanciuniv.edu).

Digital Object Identifier 10.1109/TSA.2005.848879

tree where node and extension scores corresponding to tags and labels are used. The underlying motivation is that sentences that are grammatically correct and free of recognition errors tend to be easier to parse and the corresponding scores in the parse tree are higher than those of the ungrammatical sentences containing errors generated by the speech recognizer. Even though spontaneous speech has ungrammatical constructions, we assume that the ungrammatical word sequences generated by the speech recognition engine is different from those produced in spontaneous speech. Since, the statistical parser used in this study is data driven, the parser model learns the ungrammatical sentences from the actual spontaneous utterances, as opposed to those from the speech recognition engine. The second technique is based on joint maximum entropy modeling of the word sequence and the semantic parse tree. Depending upon using shallow parsing or full parsing two maximum entropy structured language modeling techniques are used to combine semantic and lexical information sources. In this study, we use lattice based posterior probabilities as the one set of features obtained from the speech recognition component and combine with the proposed semantic features in a probabilistic framework for each word or concept. Furthermore, we also use dialog state information as an explicit additional feature to investigate its effect on the overall results.

The rest of the paper is organized as follows. In Section II, we briefly present the semantic analysis employed in our work. We describe the maximum entropy based semantic structured language models in Section III. Section IV defines the semantic confidence features followed by the feature combination from multiple sources. Experimental results are presented in Section VI. Finally, Section VII summarizes the findings.

II. Semantic Analysis

Semantic analysis involves finding the semantic units that span words or word groups and modeling the relationships among these units in a sentence. Semantic units are assigned certain tags and labels. Each word is assigned a tag associated with its semantic content. Semantically related word(s) are grouped under a semantic label. Moreover, higher level relationships among semantic unit groups are also modeled. The semantic analysis employed in this study is based on statistical classing and parsing and is currently used in limited domain dialog systems. Like many other statistical system, our statistical parser and classer require annotated training data. During annotation we impose the semantic relationships among the words and word groups in a hierarchical manner. The classer data is composed of the original sentence annotated with classes. The parser data is based on the classed sentences, and is annotated with meanings of the constituents. The decision tree based statistical classer/parser uses the training data to assign probabilities to each node and extension in a classer/parse tree [11].

The parser works in a left-to-right and bottom-up fashion. At any given step, the parser performs feature value assignment corresponding to a parser action. The parser looks for the leftmost word that needs a tag, tag that needs to be extended to a label, and then applies the model to complete the necessary action. Each parser action is assigned a probability given the current context. An example of classer and parser output is shown in Fig. 1. As seen in the figure each word is assigned a tag and certain tags are grouped under a label to form a constituent.

For the sake of simplicity we consider only the four main feature values. Let $N^k = [N_l^k \ N_w^k \ N_t^k \ N_e^k]$ refer to the 4-tuple feature values at the kth node in the current parse state. These feature values are "label" (N_l^k), "word" (N_w^k), "tag" (N_t^k), and "extension" (N_e^k). The probability distribution for each feature value is estimated using conditional models.

A tag is a leaf node in a parse tree. The tag feature value prediction is conditioned on the two words to the left, the two words to the right, and all information at the two nodes to the left and two nodes to the right [23]

$$p\left(N_t^k \,|\text{context}(t)\right) \approx p\left(N_t^k \big|\, w_i w_{i-1} w_{i-2} w_{i+1} w_{i+2} \times N^{k-1} N^{k-2} N^{k+1} N^{k+2}\right). \quad (1)$$

The label feature value prediction is conditioned on all the information in the two nodes to the left and two nodes to the right, from the two leftmost and two rightmost children of the current node

$$p\left(N_l^k \,|\text{context}(l)\right) \approx p\left(N_l^k \big|\, N^{k-1} N^{k-2} \times N^{k+1} N^{k+2} N^{c_{-2}} N^{c_{-1}} N^{c_1} N^{c_2}\right). \quad (2)$$

Extension features are the directions of the connections between two nodes. These features can take one of the four possible values: *left, right, up*, and *unary*. For example, in the constituent "los angeles california" in part A of Fig. 1, the extension feature value for the word "los" is *right*, for "angeles" *up*, and for "california" *left*. The extension feature value *unary* is reserved for the case where a tree node extends directly up to another tree node, as in the case of "fly" in the same tree. An extension probability represents the probability of placing an extension in one of the four directions between the current node and its parent node given the "context" as given in (3). Likewise, the extension feature value prediction is conditioned on the current node information that is being extended, all information from two nodes to the left and two nodes to the right, and the two leftmost and two rightmost children of the current node

$$p\left(N_e^k \,|\text{context}(e)\right) \approx p\left(N_e^k \big|\, N_w^k N_t^k N_l^k N^{k-1} \times N^{k-2} N^{k-1} N^{k+1} N^{k+2} N^{c_{-2}} N^{c_{-1}} N^{c_1} N^{c_2}\right). \quad (3)$$

We describe how some of these feature value predictions are used as semantic features for confidence measurement in Section IV-A.

The parser learns the characteristics of a sentence that lead to certain tags, labels, or extensions. Classing can be considered as shallow parsing. The classer output is used as input to the parser. Therefore, parsing in essence is a two step process. The function of the classer is to group together the words that are part of a concept. The parser takes the classer output and builds a hierarchical full semantic parse tree. The corresponding parse tree for the classer tree is given in Fig. 1. Here, semantically related concepts are grouped at a higher level. The statistical parser uses the training data to examine the training sentences to find the best combinations of sentential clues that works best across all the training sentences [10].

A parsing tree is represented as a connected, single-rooted graph with feature values at each node. The probability of a feature value assignment at a particular node in principal is conditioned on the information available at other nodes in the partially constructed tree. The parser assigns probability to a parse tree T given the sentence $S, P(T \mid S)$.

The probability of a complete parse tree (T) of a sentence (S) is the product of probabilities of feature value prediction made to build the tree. Let d_i denote a decision for a feature value assignment at the ith node. The probability of the decision is defined on the following set: $p(d_i) \in \{p(N_t^i \mid \text{context}(t)), p(N_l^i \mid \text{context}(l)), p(N_e^i \mid \text{context}(e))\}$. Each decision is conditioned on all previous decisions. The probability of a parse tree given the sentence is given below [23]

$$P(T|S) = \prod_{d_i \in T} P(d_i|d_{i-1}, d_{i-2} \ldots d_1, S) \tag{4}$$

where d_i is any allowable parser action. For example, it may be assigning a specific tag to a word, or assigning a specific label to a node in parse tree or an extension from a tag to a label or from label to a parent label. Each decision sequence defines a unique parse and the associated cumulative probability. The parser selects the one with the maximum probability.

III. Maximum Entropy Based Semantic Structured Language Modeling

The Maximum entropy (ME) method presents a framework to combine multiple overlapping information sources in an effective way. Each such knowledge source gives rise to a set of constraints that is imposed on the combined model. These constraints are typically expressed in terms of marginal distributions. The intersection of all the constraints contains a set of probability functions which are consistent with the knowledge sources or the feature expectations. Among these probability functions the ME approach chooses the function with the highest entropy. ME has been widely used in statistical language modeling [17], [22]. Because of the convenience of combining multiple information sources, ME has also been used for syntactic structured language modeling [20] and semantic structural language modeling [19]. ME modeling matches the feature expectations exactly while making as few assumptions as possible in the model. ME modeling presents a unified framework to combine lexical and semantic content and the structure of the sentence in an effective manner. Therefore, we can assign a joint probability to each word based on its lexical and the semantic history. We have used a classer and a parser to extract semantic content of a sentence.

In the ME framework we can view each information source as defining one or more subsets of the event space (w, h), where w is a word and h is the history associated with w. We associate a constraint with each subset to satisfy a certain statistic of the training data over that subset of the event space. The ME method combines the multiple information sources in the following way:

$$P(w \mid h) = \frac{e^{\sum_i \lambda_i f_i(w,h)}}{\sum_{w'} e^{\sum_i \lambda_i f_i(w',h)}} \tag{5}$$

where w is the current word, f_i are the feature constraint functions (or indicators) that are activated for a certain history and defined as

$$f_i(w,h) = \begin{cases} 1, & \text{if } w = w_i \text{ and } h \in S \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where h represents the history which may include previous words as well as tags and labels that can be used in predicting the current word. Here S is a subset of the event space. In [1], we used ME to model sentence based syntactic and semantic information. Semantic information is obtained from the semantic classer and parse trees. We computed the joint probability of a word sequence and a parse tree: $P(W, C)$ [1]. The first step in building the Maximum Entropy model is to represent a classer/parse tree as a sequence of words, tags, and labels. The labels are divided as begin-label and end-label.

Basically, this representation (an example is given in Section IV-A along with the token probabilities) is equivalent to enriching the original text with tags and labels. This representation allows us to define the boundaries for the semantic constituents and take long range semantic information into account. Each token is an outcome of the joint model. Since the tags are already included in the classes used in the class-based language modeling, we ignored them in our analysis. In [1], we proposed a set of maximum entropy based structured language modeling (MELM) techniques: MELM1, MELM2, and MELM3.

MELM1 is the ME counterpart of an ordinary n-gram language model. MELM2 and MELM3 include semantic information besides lexical information. MELM2 and MELM3 differ in the level of semantic analysis employed. From now on we will use MELM's when we refer to both MELM2 and MELM3. Interpolating MELM's with the class-based trigram provided significant improvement over a sophisticated class-based language model [1]. These improvement are due to the inclusion of new semantic information that was not part of the original speech recognition system.

IV. Semantic Confidence Features

The issue that we want to address is what are the best features that can be obtained from semantic analysis. These features should be statistically based to combine with the probabilistic features from the speech recognition component. In previous studies the semantic information is mainly modeled in the form of slot/concept sequences, the coverage of the words by the parser, and the features derived from these forms [3], [8], [2], [14], [12]. However, some of the scores attributed to words are *ad hoc* and there is not a clear interaction in the modeling between lexical and semantic content of the sentence. Our approach differs from these studies in the following ways: the amount of semantic information used, tightness of the integration of lexical and semantic information and statistical modeling of lexical and semantic information. We propose two techniques which have these properties. The first technique is based directly on the classer or parse tree. The second technique is based on Maximum Entropy based statistical modeling of lexical and semantic information obtained from the classer/parse tree. Next, we discuss the first technique and how the semantic features are obtained.

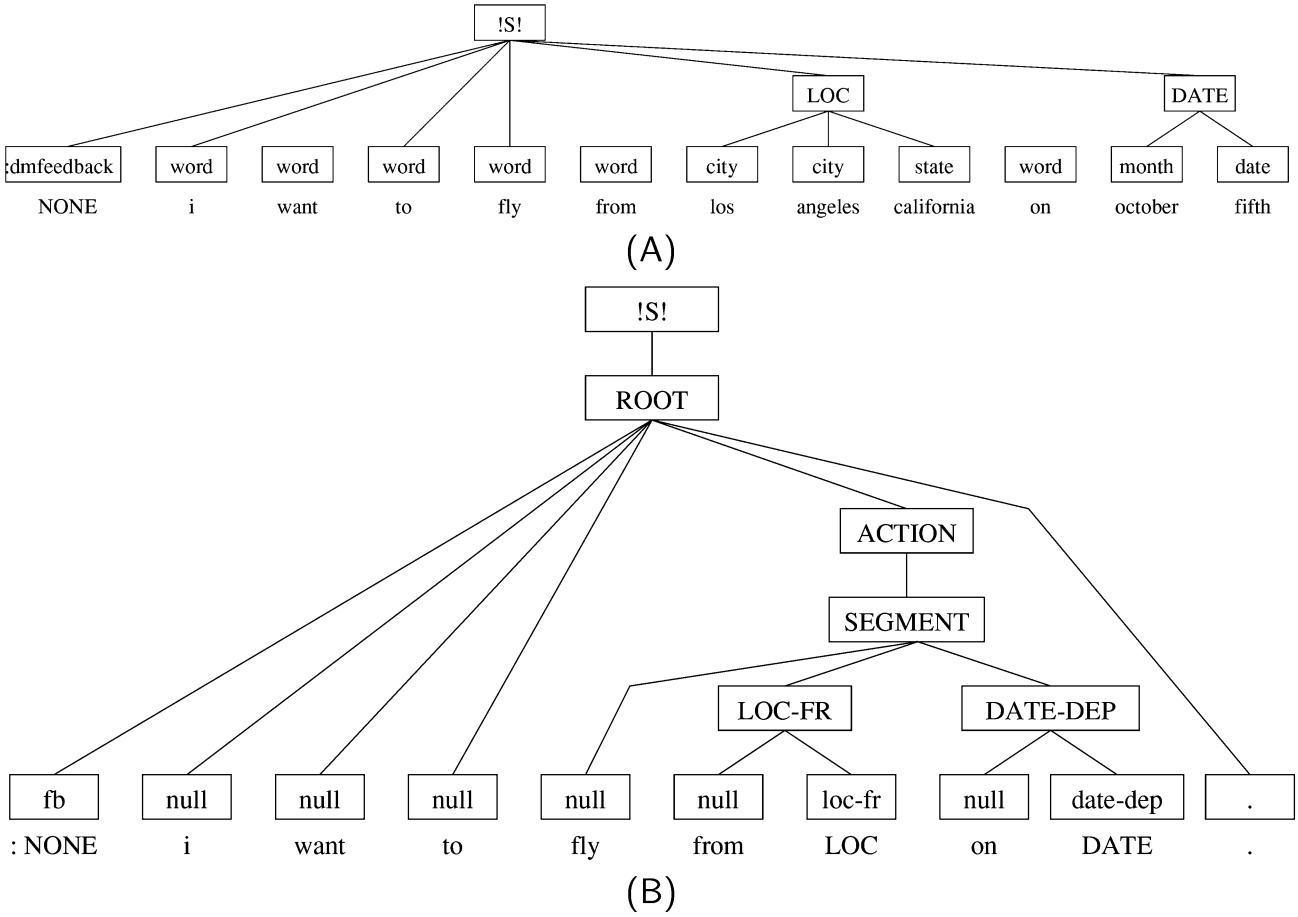


Fig. 1. Classer and Parser outputs for an example sentence. (a) Eexample of a semantic classer output. (b) Parser output for the same example.

A. Semantic Tags, Labels, and Extensions

The classer/parser performs a left-to-right bottom-up search to find the best parse tree for a given sentence. During search, each tag node, label node, and extension in the parse tree is assigned a probability. The node probability represents the probability of assigning a feature value to it given the "context" as given by (1) and (2). Similarly, an extension probability represents the probability of placing an extension in a certain way between the current node and its parent given the "context" as given in (3). When the parser is conducting the search both lexical and semantic clues are used to generate the best parser action. The degree of confidence while assigning the tag and label feature values is reflected in the associated probabilities. If word does not "fit" in the current lexical and semantic context, its tag and label are likely to be assigned low probabilities. Therefore, using these probabilities as features in the confidence measurement is a viable way to capture the semantics of a sentence.

In our analysis we extracted features from both the classer and parse trees. For word level confidence measurement we considered tag and extension features from both trees. The following set of statistical semantic features are extracted:

$$\begin{aligned}\mathbf{cTag} &= p(N_t \,|\, \text{context}(t), T_c)\\ \mathbf{cTagExt} &= p(N_e \,|\, \text{context}(e), T_c)\\ \mathbf{pTag} &= p_(N_t \,|\, \text{context}(t), T_p)\\ \mathbf{pTagExt} &= p(N_e \,|\, \text{context}(e), T_p)\end{aligned} \tag{7}$$

where T_c and T_p denote classer and the parse trees, respectively. Here **cTag** and **cTagExt** denote the classer tag and tag extension probabilities, respectively. Likewise **pTag** and **pTagExt** denote the parser tag and tag extension probabilities, respectively. The example in Table I shows the classer tree given in Fig. 1 in text format along with the node and extension probabilities. This sentence is from the DARPA Communicator air travel reservation domain. In this domain the number of tags is proportional to the number of concepts in the task. All the words belonging to a concept are tagged with a related tag. All the remaining words that are not part of a concept are tagged as **word**. There is even smaller number of labels for the classer. These labels include **LOC, DATE, TIME, NUMFLT, AIR, PRICE, CLASS**.

Each term in the representation given in Table I is called a token. Note that each token is assigned a pair of probabilities. The first probability is for the node and the second probability is for the extension. In the table, "0.960 17" is the classer tag probability (**cTag**), and "0.995 701" is the classer tag extension probability (**cTagExt**) probability for the word "california." Similarly, the corresponding parser tag (**pTag**) and parser tag extension (**pTagExt**) probabilities are extracted from the parse tree. In the beginning of the sentence ":NONE" indicates the dialog state for the sentence. The probability of the classer tree given in Table I is 0.851 88, which is the probability of the best classer tree among all the possible set of trees that can be assigned to this sentence. This probability is obtained according

TABLE I
NODE AND EXTENSION PROBABILITIES FOR THE CLASSER TREE

Classer Tree Probabilities						
	[!S!	:NONE_dmfeedback	i_word	want_word	to_word	fly_word
Node	1	1	0.994147	0.999605	0.998604	0.999605
Extension	1	0.997937	0.995734	0.995734	0.995734	0.995734
	from_word	[LOC	los_city	angeles_city	**california_state**	LOC]
Node	0.99371	0.999976	0.997406	0.998609	**0.96017**	0.999976
Extension	0.995734	0.999852	0.999775	0.957721	**0.995701**	0.999852
	on_word	[DATE	october_month	fifth_date	DATE]	!S!]
Node	0.99948	1	0.999516	0.984234	1	1
Extension	0.995734	0.998174	0.997417	0.995631	0.998174	1

TABLE II
NODE AND EXTENSION PROBABILITIES FOR THE PARSE TREE

Parse Tree Probabilities						
	[!S!	[ROOT	:NONE_fb	i_null	want_null	to_null
Node	1	0.999516	1	0.984263	0.988049	0.994018
Extension	1	0.999522	0.984119	0.992231	0.995098	0.995098
	[ACTION	[SEGMENT	fly_flights	[LOC-FR	from_null	LOC_loc-fr]
Node	0.997064	1	0.990373	1	0.702338	0.991334
Extension	0.995984	0.995123	1	0.998472	0.999981	0.994569
	LOC-FR]	[DATE-DEP	on_null	DATE_date-dep	DATE-DEP]	SEGMENT]
Node	1	1	0.544135	0.998505	1	1
Extension	0.998472	0.990492	0.974586	0.999671	0.990492	0.995123
	ACTION]	._.	ROOT]	!S!]		
Node	0.997064	1	0.999516	1		
Extension	0.995984	0.999519	0.999522	1		

to (4) by multiplying all the node and extension probabilities given in the Table I. Potentially, any of the probabilities in this tree can be used as a semantic feature. For the classer tree we decided to use only the tag and tag extension probabilities. Note that classer labels become tags when classer outputs are fed into a parser. The corresponding parser output for the same sentence is given in Table II in text format.

In Table II, words that are not part of any concept are assigned the **null** tag. The parser performs detailed semantic analysis. For example words grouped under **LOC** are distinguished as **LOC-FR** (departure location) or **LOC-TO** (destination location). Moreover such semantically related concepts as **LOC-FR** and **DATE-DEP** (departure date) are grouped under **SEGMENT**. Likewise, we use tags and tag extension probabilities as our confidence features at the parser level. Our statistical parser requires a end-of-sentence marker (a ".")[1]

The classer assigns a pair of probabilities to each word. However, the parser assigns a pair of probability to classer concepts and the remaining words. **LOC-FR** and **DATE-DEP** given in the parse tree shown above are examples of classer concepts. The parser probabilities for the words are derived in the following way. Assume that a sentence, S has the following word sequence: $S = [w_1, w_2, \ldots, w_T]$. The classer chunks the word sequence and generates a concept sequence: $C = [c_1, c_2, \ldots, c_{T'}]$, where $T' \leq T$, for the parser input. The $\mathbf{pTag}_{w_i}$ and $\mathbf{pTagExt}_{w_i}$ probabilities are assigned to word w_i according to the following relationship:

$$\begin{aligned} \mathbf{pTag}_{w_i} &= \mathbf{pTag}_{c_j}, \quad \text{if } w_i \in c_j \\ \mathbf{pTagExt}_{w_i} &= \mathbf{pTagExt}_{c_j}, \quad \text{if } w_i \in c_j \end{aligned} \tag{8}$$

[1]That is the reason why the sentence is augmented with a period at the end although it was not part of the original sentence.

where $\mathbf{pTag}_{c_j}$ and $\mathbf{pTagExt}_{c_j}$ are the concept tag and tag extension probabilities, respectively. These probabilities are assigned by the parser. Therefore the 4-tuple probabilities extracted from the semantic analysis for word w_i is $[\mathbf{cTag}_{w_i}\ \mathbf{cTagExt}_{w_i}\ \mathbf{pTag}_{w_i}\ \mathbf{pTagExt}_{w_i}]$.

B. MELM2 Features

MELM2 is one of the language models proposed in [1] and employed seven types of questions about the current token in a sentence [1]. In addition to regular n-gram questions for trigrams, four more questions are used regarding the semantic structure of the sentence. MELM2 uses only the semantic classer tree. Therefore questions regarding the semantic content of the sentence are limited to classer labels and tags. The set of semantic questions used for MELM2 modeling is given as follows:

- current active parent (L_i);
- L_i and number of words to the left since starting the current constituent, N_i;
- L_i, N_i, and previous word token, w_{i-1};
- the previous completed constituent (O_i) and number of words to the left since completing O_i, M_i.

The history given in (5) above consists of answers to n-gram and these questions. The language model score for a given word in MELM2 model is conditioned not only on the previous words but also tags, labels, and relative coverage of these labels over words. Therefore the MELM2 probability can be computed as follows [1]:

$$P_{\text{MELM2}}(\tau_i \,|\, h_i) = \frac{\exp(\sum_k \lambda_k f_k(y_i, h_i))}{Z(h_i)} \tag{9}$$

where $h_i = [u_i, w_{i-1}, w_{i-2}, L_i, N_i, O_i, M_i]$ and $Z(h_i)$ is a history dependent normalization term. Note that u_i is the unigram

history for the current token τ_i. Note that MELM2 is defined on the token sequence. MELM2 presents an effective statistical method to combine word sequences with the semantic classer tree. Therefore we can use the MELM2 score as a feature for confidence measurement. However, MELM2 for a given word only depends on the previous word sequence and the parse tree up to that word.

In [6], it is observed that on a subset of the Switchboard development test data correctness on the current word has a significant effect on the correctness of the next word. For example, the next word is correct 87% of the time when the current word is correct and only 48% of the time when it is incorrect. Even though it is a different data set, this observation suggests that we can expect a low score for the current word if the previous word is recognized incorrectly. We can incorporate the context information into confidence measurement as follows.

Assume that a sentence is represented by the following token sequence, $[\tau_1, \tau_2, \ldots, \tau_M]$ with the actual word sequence, $[w_1, w_2, \ldots, w_T]$. Since there is always a begin ([!S!) and end (!S!]) root label for any sentence, $T \leq M - 2$. Let N be the size of the history of the current token. Then, we define an augmented token sequence as $[\hat{\tau}_1, \hat{\tau}_2, \ldots, \hat{\tau}_{M'}]$ with the following constraint:

$$\hat{\tau}_j = \begin{cases} \tau_1, & \text{if } 1 \leq j \leq N \\ \tau_M, & \text{if } (M+N) < j \leq (M+2N) \\ \tau_{j-N}, & \text{if } N < j \leq (M+N) \end{cases} \tag{10}$$

where $M' = M + 2N$. We define the token context around the current word, $w_i = x_0$, by the $(2\mathrm{N}-1)$-tuple $[x_{-N}, x_{1-N}, \ldots, x_{-1}, x_0, x_1, \ldots, x_N]$. Assuming an equal context size on both side of a word, the $(2\mathrm{N}-1)$-tuple score representing a word is formed with the following relationship:

$$\begin{aligned} x_0 &= \hat{\tau}_j \\ x_i &= \hat{\tau}_{j-i} \end{aligned} \tag{11}$$

where $-N \leq i \leq N$ and $2 \leq j \leq M'-1$. For our experiments we considered $N = 1$ and $N = 2$. When N is set to 1, besides the **MELM2** score for the current word $w_i = \hat{x}_0$, we consider the scores of the left and the right neighbors of w_i leading to triple scores: $[x_{-1}\ w_i\ x_{+1}]$. This feature is named **MELM2-ctx3** implying a context of three. Similarly when $N = 2$ we consider two neighbors to the left and two neighbors to the right of the current word leading to the 5-tuple $[x_{-2}\ x_{-1}\ w_i\ x_{+1}x_{+2}]$. This feature is named accordingly **MELM2-ctx5**. In our experiments we also considered only words rather than tokens as the neighbors, however the performance was inferior to those of having tokens as the neighbors.

C. MELM3 Features

MELM3 combines semantic classer and parser and uses a full parse tree [19]. The full parse tree presents a complete semantic structure of the sentence where in addition to classer information semantic relationships between the constituents are also derived. The following features are used to train a Maximum Entropy based statistical model:

- unigram history, u_i;
- previous word, w_{i-1};
- two previous words, w_{i-1} and w_{i-2};
- current parent label, L_i and the number of words to the left since the start of the constituent, N_i;
- L_i, and the current grandparent label, G_i;
- the most recent completed constituent, O_i and number of words to the left since completing O_i, M_i.

Although the trees that the questions are based are different, MELM2, and MELM3 share similar questions. Indeed, only the fifth question of MELM3 is not included in the MELM2 question set. Note that even though these specific question sets are selected for MELM2 and MELM3, any question based on classer and parse trees can be a legitimate choice in Maximum Entropy modeling. We experimentally determined these question sets to be performing well. Inclusion of additional questions did not improve the performance for the current task. Similarly, we can take the context into account by defining an array of scores centered on the current score. We consider **MELM3-ctx3** and **MELM3-ctx5** as the counterparts of **MELM2-ctx3** and **MELM2-ctx5**, respectively.

D. Posterior Probability Computation

The posterior probabilities are obtained from the sausages [13], A sausage is a simplified word graph with a specific topology. Generation of sausages from the word graphs is motivated by minimizing the word error rate rather than sentence error rate. The standard speech recognition objective function is based on maximizing the posterior probability of the word sequence W given the A acoustic information, $P(W \mid A)$. However, the performance metric, word error rate (WER) is an edit distance between a hypothesis W and the reference string R. The new objective to minimize is the expected WER given the posterior distribution [13]

$$E_{P(R \mid A)} WE(W,R) = \sum_R P(R \mid A) WE(W,R). \tag{12}$$

A practical solution to this new hypothesis selection method is proposed in the form of sausages. The word graph is converted into a sequence of confusion sets along time. Each confusion set consists of a group of words, which are competing hypotheses for a certain time interval. The posterior probabilities for each word is obtained by summing the probabilities of all the paths containing that word. In each confusion set the sum of posterior probabilities is one. Parts of a sausage for the example sentence used so far are shown in Fig. 2.

As seen in the figure there are several alternative word hypothesis in each bin providing a compact representation of the competing hypothesis generated by the speech recognition engine. Here, $\langle \mathrm{eps} \rangle$ refers to "epsilon" transition, which provides costless transitions across bins. The name "sausage" comes from the fact the graph in Fig. 2 resembles a sausage in its literal sense.

V. Feature Combination

The speech recognition acoustic and language model vocabularies contain compound words that help to improve the system performance. The semantic analysis step on the other hand does not contain any compound words. It even splits such words as $\langle \mathrm{NEW\_YORK} \rangle$ into $\langle \mathrm{NEW} \rangle$ and $\langle \mathrm{YORK} \rangle$ and associates a pair of probabilities with each of the component words. The

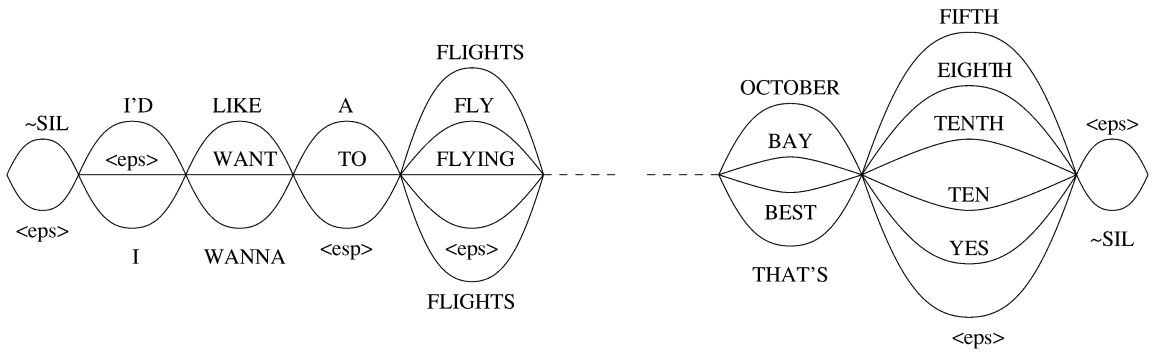


Fig. 2. Sausage generated for an example sentence.

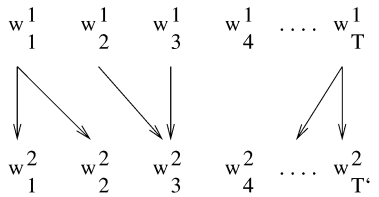


Fig. 3. Word alignments between two word sequences.

word unit is also different for the MELM2 based features. The issue here is what to use as a word unit and how to associate probabilities coming from different sources with these words.

Assume that S_1 and S_2 are two realizations of the same sentence S differing in the compounding of the word sequence. Let $S_1 = [w_1^1, w_2^1, \ldots, w_T^1]$ and $S_2 = [w_1^2, w_2^2, \ldots, w_{T'}^2]$ be the word sequences generated by the first and the second sources, respectively. Note that any of the words can be a set of words (compound word) in both sequences. The alignment between these sequences can be a general one as shown in Fig. 3.

First we determine the alignment between S_1 and S_2. We generate a combined word sequence using the alignments between the sentences via the following relationship:

$$w_k = \begin{cases} w_i^1, & \forall i \text{ such that } w_i^1 \subset w_j^2 \\ w_j^2, & \forall j \text{ such that } w_j^2 \subset w_i^1 \end{cases} \tag{13}$$

where w_i^1 and w_j^2 are aligned to each other. The probabilities belonging to w_k's are inherited from the probabilities of the aligned segments. For example, for the first line of the equation given above w_k is assigned the probabilities corresponding to w_i^1 as well as w_j^2. Note that in this case w_k is assigned the same w_j^2 probability for each w_i^1 probability. The same method can be extended to more than two feature streams.

VI. Experimental Results and Discussion

We have carried out experimental investigations of confidence measurement on the IBM DARPA communicator system [9]. MELM's (both MELM2 and MELM3) and baseline class based language models are trained on 137 K sentences in the air travel domain. An additional 18 K sentences are used for smoothing of the class based language model. The MELM's are trained using the improved iterative scaling algorithm using fuzzy smoothing [1], [17]. The confidence measurement training data is formed by pooling eight other DARPA communicator sites' evaluation data. This data was from the calls received by those Communicator systems during the National Institute of Standards (NIST) June 2000 evaluations. The corresponding evaluation data for the IBM DARPA Communicator system is used as test data. Note that many of these communicator sites have different dialog strategies. Although the task is the same, the dialog questions and the answers can be quite different. Having no overlap within the training and test data as far as the systems are concerned adds one more degree of difficulty to our experiments. The confidence training data consist of 10 640 sentences and 28 666 words. The test data consist of 1173 sentences and around 3600 words. Therefore an average sentence contains about 2.7 words. The acoustic models are trained using air travel and generic telephony data. A separate class-based trigram language model with deleted interpolation is trained on the MELM's training and held-out data and used during speech recognition.

For each sentence in the confidence training and test data, a sausage is generated and the consensus hypothesis, which is the best path from the sausage, is hypothesized as the speech recognition output. The best path computed based on the posterior probability resulted an average 1.4% improvement over the confidence measurement training and test data compared to simple Viterbi-based decoding (21.1% versus 19.7%). This is consistent with the results obtained on other tasks [13]. Each word/concept is labeled as correct ("1") or incorrect ("0") after aligning the hypothesis with the reference transcript. All the recognition hypotheses are classed using statistical semantic classer and parser. Each sentence is scored with MELM's to assign semantic probabilities to each word. The corresponding semantic features are extracted for all the words in the sentence. All of the positive (correct recognition) and negative (misrecognition) examples are pooled in two sets.

There are a number of studies comparing such classifiers as Neural Networks, Decision Trees and Support Vector Machines (SVM's) [2], [8]. Depending on the particular application a specific classifier can marginally outperform others with the tuning of its parameters. However, results in general indicate that these classifiers perform similarly [2], [8]. In our experiment we used a decision tree algorithm to classify the positive and negative examples. The decision tree has used the raw scores of each feature. In our decision tree algorithm, the tree is grown by partitioning the data recursively in each node until either the node becomes homogeneous or the node contains too few observations (≤ 200). We have used other minimum observation count

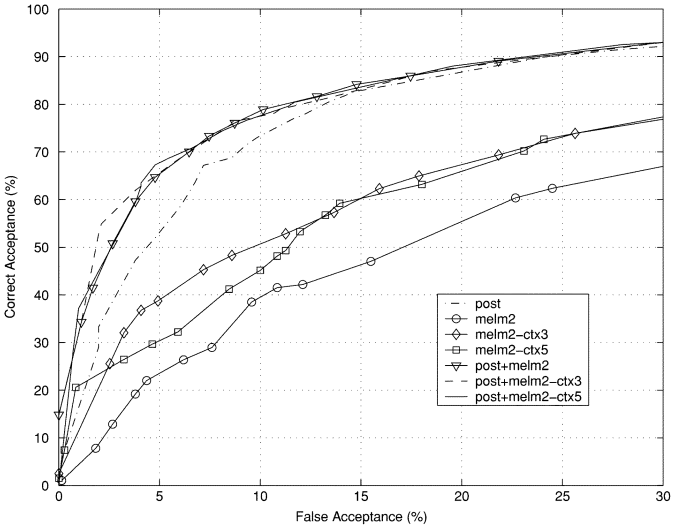


Fig. 4. Word level ROC for combination of posterior probability with MELM2 based features.

thresholds to maximize the performance for all the experiments. However, the performance did not change significantly as long as the threshold is set in the range of 50–500. In order to predict the correctness of a word from the features, one follows the path from the root, to a leaf, according to splits at the interior nodes.

It is useful to have a single measure of performance for confidence measurement. The Equal Error Rate (EER) is one such measure. EER is the operating point on an Receiver Operating Characteristic (ROC) curve where False Acceptance (FA) is equal to False Rejection (FR). However, for spoken dialog systems it is not a useful operating point as one needs to accept as many correct words as possible at a very low False Acceptance (FA) rate. Our observation of user behavior with this and similar systems led us to believe that FA rates of 5–10% is the likely operating range for dialog systems. The FA and CA are calculated using the following formula:

$$\mathrm{FA} = \frac{\#\,\text{of falsely accepted words}}{\text{Total}\,\#\,\text{of negative examples}} \times 100$$
$$\mathrm{CA} = \frac{\#\,\text{of correctly accepted words}}{\text{Total}\,\#\,\text{of positive examples}} \times 100. \qquad (14)$$

Next, we conducted three sets of experiments. The first set measures the confidence using lattice based posterior probability as baseline and the proposed classer/parser and MELM based features at the word level. In the second set, the effect of including dialog state (DS) explicitly on the confidence measurement is investigated. The last set of experiments measure the confidence at the concept level.

A. Word Level Confidence Measurement

Word level confidence measurement refers to associating a score or probability with each word. In Fig. 4, we present the ROC curves for MELM2 based features. Here, **MELM2** refers to the language model score for a given word, and **MELM2-ctx3** refers to **MELM2** score of context three where previous and the next scores are included as part of the current score. Similarly, **MELM2-ctx5** refers to a window of five scores around the current score. Including context around the current word improves the MELM2 performance. For example, **MELM2-ctx3** outperforms **MELM2** by 16% and 6% at 5% and 10% FA rates, respectively. Note that in general **MELM2-ctx5** does not perform as well as **MELM2-ctx3**. We attribute this to short sentences (average of 2.7 words per sentence). In applications where the average word count in a sentence is large including context information may be beneficial. Combining MELM2 based features with the posterior probability improves the CA rate significantly. For example at 5% FA rate the CA rate for **MELM2-ctx5** combined with the posterior probability is 14.6% better than posterior probability alone. Note that the most interesting part of the ROC curves for dialog systems is between 5–10% FA rate, and the feature combination is particularly effective in this range. Although the performance based on individual MELM2 features alone is fairly low compared to posterior probability alone, when combined with the posterior probability the overall result is improved. MELM2 based features clearly add complementary new information to posterior probability information. We extracted the CA rates at 5% and 10% FA rate from the ROC curve and presented them in Table III. The best improvement at 5% FA is 14.6% when posterior probabilities are combined with **MELM2-ctx5**.

TABLE III
CORRECT ACCEPTANCE (CA) RATES AT 5% AND 10% FALSE ACCEPTANCE (FA) RATES FOR MELM2 BASED FEATURES

Performance of the MELM2 Based Features.(%)		
	5% FA	10% FA
Posterior	53.1	73.4
MELM2	23.5	39.5
MELM2-ctx3	39.0	50.7
MELM2-ctx5	30.4	45.1
Posterior + MELM2	65.4	78.6
Posterior + MELM2-ctx3	65.6	77.5
Posterior + MELM2-ctx5	67.7	77.6

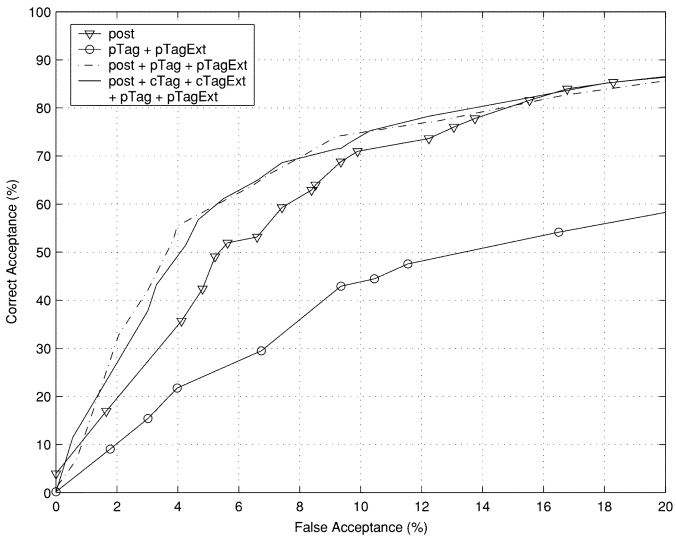


Fig. 5. Word level ROC for posterior probability, classer, and parser based features.

The results for the classer/parser based features are shown in Fig. 5. The features considered here are **cTag, cTagExt, pTag**, and **pTagExt**. Although there are a number of combinations of these features among themselves and with the posterior probability, not all of them are included in Fig. 5. The performance

TABLE IV
CORRECT ACCEPTANCE (CA) RATES AT 5% AND 10% FALSE ACCEPTANCE (FA) RATES FOR CLASSER/PARSER BASED FEATURES

Performance of the Classer/Parser Features.(%)		
	5% FA	10% FA
Posterior (Post)	45.7	71.0
cTag	17.4	35.2
cTag + cTagExt	20.1	37.9
pTag	16.8	34.2
pTag + pTagExt	24.6	43.8
Post + cTag	54.5	70.9
Post + cTag + cTagExt	55.3	71.3
Post + pTag	52.9	73.2
Post + pTag + pTagExt	58.9	74.9
Post + cTag + pTag + pTagExt	54.9	71.9
Post + cTag + cTagExt + pTag + pTagExt	58.5	74.1

of the some of the remaining combination are given in Table IV. Even though the relative improvement of these features when combined with posterior probability is similar, the best performance is obtained when posterior is combined with **pTag** and **pTagExt**. At 5% FA rate they outperformed posterior probability by 13%.

At high FA rates room for improvement shrinks rapidly. The improvement in CA for both feature sets at 10% FA rate is moderate (4–5%). Note that the posterior probability has different CA rates at the same FA rates in Tables III and IV. This is because of the fact that some of the compound words used by classer/parser and **MELM2** are different. For example, "BUFFALO_NEW_YORK" is a single unit for posterior probability and MELM2 but it is three units: "BUFFALO," "NEW," and "YORK" for classer/parser. As explained in Section 5, the classer/parser assigns the tag and extension probability to each of these words. Therefore the same posterior score is repeated three times when combined with the classer/parser scores. As a result total number of positive and negative examples are different for MELM2 and classer/parser based feature sets that lead to different ROC for the posterior probability. Therefore ROC curves in each figure should be considered only within that figure rather than across figures.

Next, we compared MELM3 based features with the posterior probability. The MELM3 based features are the counterparts of MELM2 based features. Since **MELM2** and **MELM2-ctx3** marginally outperformed **MELM2-ctx5** when combined with the posterior probability, we combined **MELM3** and **MELM3-ctx3** features with the posterior probability. The ROC curves are plotted in Fig. 6. Combining these features with the posterior probability improved the CA rate significantly when the FA rate is less than 10%. The CA rates at 5% and 10% FA rates are given in Table V. The improvement is similar to those of the MELM2 based features. Although in [19], we determined that MELM3 outperformed MELM2 for certain data sets, we do not have any evidence that MELM3 outperforms MELM2 (or vice-verse) for confidence measurement. Therefore, next we combine the dialog state information explicitly with only MELM2 based features and the posterior probability.

B. Word Level Confidence Measurement With Dialog State

It has been shown that including dialog state or speaker turn information in language modeling improves the speech recognition performance [7], [24]. As given in the example in Section IV-A every sentence in the confidence measurement training and test data is tagged with a dialog state. Therefore the parser and the classer based features use dialog state information. On the other hand, neither the MELM or class-based trigram language model training data have information regarding the dialog state. Therefore the MELM based scores are not based on the dialog state information. However, the decision tree building algorithm allows one to combine multiple information sources. We combined MELM2 based features with the dialog state information by including it as an additional feature. Although, dialog state information is used to compute the classer and parser features, we wanted to use it explicitly as an additional feature in the decision tree building process. The dialog states or the parser feedback tags are **DATE, DONE, NONE, LIST, TIME, LOC**. A list of the parser feedback tags (dialog states) with their frequency of occurrence is given in Table VI.

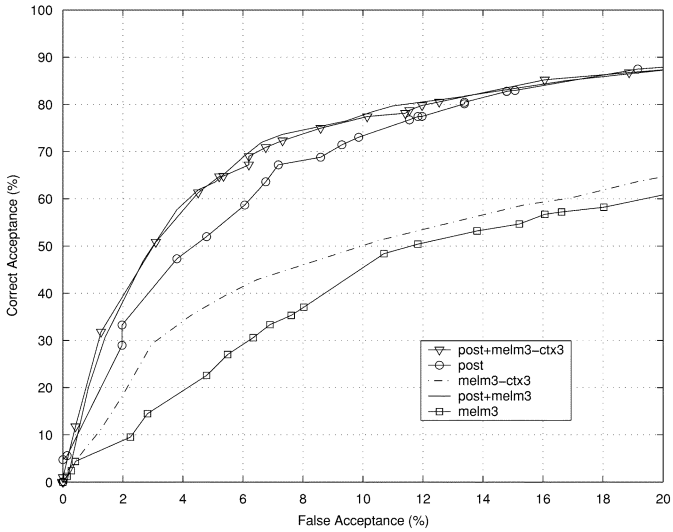


Fig. 6. Word level ROC for posterior probability, and MELM3 based features.

TABLE V
CORRECT ACCEPTANCE (CA) RATES AT 5% AND 10% FALSE ACCEPTANCE (FA) RATES FOR MELM3 BASED FEATURES

Performance of the MELM3 Based Features.(%)		
	5% FA	10% FA
Posterior	53.1	73.4
MELM3	23.9	44.1
MELM3-ctx3	38.0	50.1
Posterior + MELM3	63.3	77.7
Posterior + MELM3-ctx3	63.7	77.2

TABLE VI
DESCRIPTION OF THE CONFIDENCE MEASUREMENT TRAINING DATA WITH RESPECT TO DIALOG STATES

Dialog State	# of Sentences	Expected Input
DATE	1212	date
DONE	1710	confirmation
LIST	39	choice from list
NONE	5393	any other input
TIME	946	time
LOC	1341	depart/destin. loc.

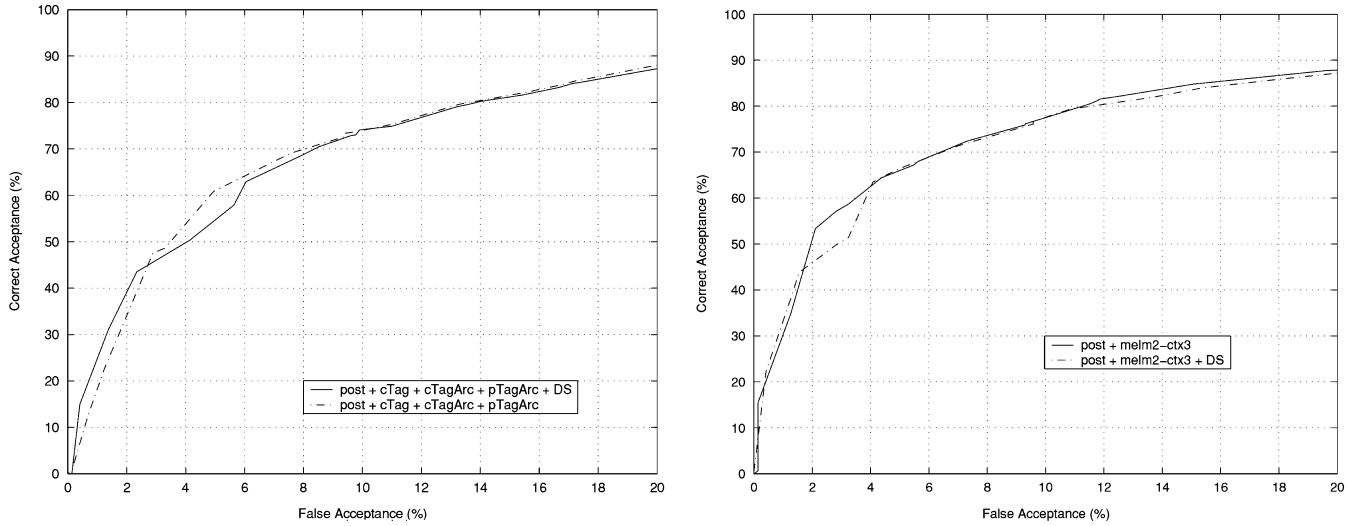


Fig. 7. Effect of DS feature on classer/parser and MELM2 based features.

The ROC curves showing the effect of Dialog State (DS) as an additional feature is shown in Fig. 7. Based on the figure DS information did not improve the performance of the classer/parser based features as well as MELM2 based features. This may be due to very few dialog states (only six). Although we have detailed dialog states we wanted to avoid fragmenting already small amount of training data (around 10.6 K sentence).

C. Concept Level Confidence Measurement

In spoken dialog systems it is not important if each and every word is recognized correctly. However, it is important if the words that are part of a concept are recognized correctly. Therefore, confidence measurement at the concept level is particularly important for spoken dialog systems. The concept is defined as the combination of word(s) belonging to a label in a classer tree. Our natural language understanding (NLU) system attempts to capture a limited number of concepts from the recognition hypothesis. The classer covers 19 concepts. A concept can span just a single word as well as a sequence of words. The semantic classer in essence finds the concepts in a sentence by grouping words that form a concept. We have 17 concepts: {AIRLINE, CLASS, AIRCRAFT, DATE, DATE-TIME, DURATION, LOC, MEAL, NAME, NUM-FLT, NUM-PASSENGER, PRICE, RT-OW, RULE, STOPS, TIME, TRANS}.

In order to assign posterior probability based confidence to the concepts, the posterior probabilities corresponding to constituent words are multiplied as

$$\gamma_{c_j} = \prod_{w_i \in c_j} \gamma_{w_i} \tag{15}$$

where c_j is a concept, γ_{w_i} and γ_{c_j} are the posterior probability of word w_i and c_j, respectively. There are several ways to obtain a concept score from the classer. For example, one can multiply all the probabilities between begin-label and end-label: **classer**. We can also normalize the final overall probability with the number of words spanned by the classer concept: **norm-classer**. Both of these scores can be expressed in the following equation:

$$p_{c_j} = \left[\prod_{i=1}^{N} p_{\tau_i}\right]^{1/M} \quad \text{such that } \tau_i \in c_j \tag{16}$$

where p_{c_j} is the concept score, τ_i is a token as defined in Section 4.2 and N is the number of tokens spanned by concept c_j. If $M = 1$ and τ_i's are constrained to be just words, we obtain a **classer** score. If $M = N$ and τ_i's are either word or label, we obtain a **norm-classer** score. Furthermore, we can combine classer concept probability with the corresponding parser **pTag** and **pTagExt** probabilities.

Extracting the concepts from the confidence training data resulted in 6855 concepts. The concept count for the test data is 919. All the words that are not part of any concept are eliminated. The decision tree is rebuilt using this concept training data and tested on the corresponding concept test data. Among all these possible alternatives combining posterior probability with the **classer** gave the best performance. In Fig. 8, we present the ROC curve for posterior probability, classer concept probability (**classer**) and their combination. ROC curves show that at the same False Acceptance (FA) level the Correct Acceptance (CA) rates are lower compared to word level confidence measurement. This is due to bias toward misrecognition. For example if any of the words is misrecognized in ⟨WEDNESDAY_NOVEMBER_FIRST_TWO_THOUSAND⟩ the **DATE** concept is assumed to be misrecognized. The ROC curves for concept level confidence measurement are not as smooth as those of the word level confidence measurement. This is mainly due to limited amount of concept test data. However, combination of posterior probability with the **classer** improved results significantly compared to that of posterior probability alone. The ROC curves are sampled at 5% and 10% FA rates and presented in Table VII. The improvement surpasses all the improvements obtained at the word level confidence measurement. For example at 5% FA rate a 20% absolute improvement (61% relative improvement)

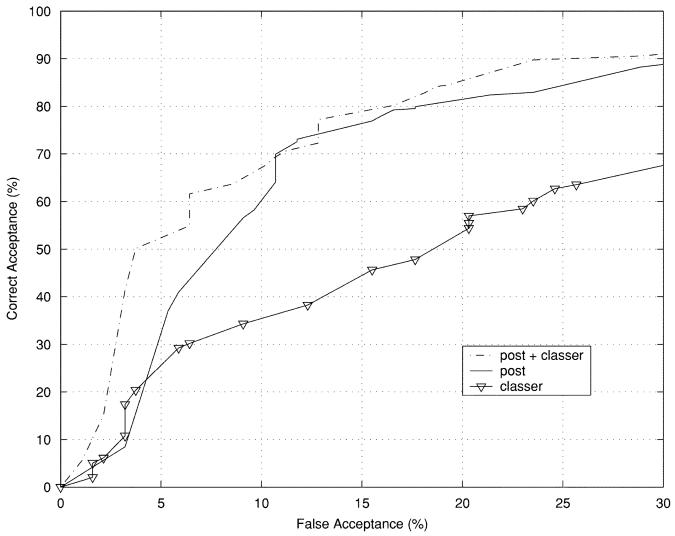


Fig. 8. Concept level ROC for posterior probability versus classer probability.

TABLE VII
CONCEPT LEVEL CORRECT ACCEPTANCE (CA) RATES AT 5% AND 10% FALSE ACCEPTANCE (FA) RATES FOR POSTERIOR AND CLASSER BASED FEATURES

Performance of the Classer/Parser Based Features at the Concept Level.(%)		
	5% FA	10% FA
Posterior	32.4	60.2
Classer	25.6	35.4
Posterior + Classer	52.4	67.1

in CA rate over posterior probability is obtained. Intuitively, this is a very satisfying result, as we expect to see greater contribution of the semantically based features when combined with the posterior probability at the concept level.

VII. CONCLUSIONS

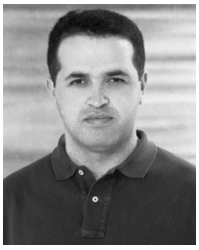
We proposed two methods to generate word level semantic features and integrate them with a lattice based posterior probability feature in a principled manner. The first set of semantic features consists of tag and tag extension probabilities for statistical classer and parse trees. The second set of semantic features are derived from the maximum entropy based semantic structured language models (MELM2 and MELM3) with variable context around a given word. The semantic features brought complementary information to speech recognition information summarized by the posterior probability. For the word level confidence measurement combination of these features with posterior probability provided an absolute improvement of around 13–14% for Correct Acceptance at 5% False Acceptance rate over posterior probability. Classer concept probabilities are combined with the posterior probability at the concept level. Compared to baseline, at 5% FA rate semantic features gave a 20% absolute improvement in CA rate. In the future, we plan to apply the proposed methods to additional domains to further demonstrate the validity of the approach.

ACKNOWLEDGMENT

The authors thank L. Mangu, R. San-Segunda, and the reviewers for valuable suggestions.

REFERENCES

[1] H. Erdogan, R. Sarikaya, Y. Gao, and M. Picheny, "Semantic structured language models," in *Proc. Int. Conf. Spoken Language Processing*, Denver, CO, Sep. 2002.
[2] R. San-Segundo, B. Pellom, K. Hacioglu, and W. Ward, "Confidence measures for spoken dialog systems," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 2001, pp. 393–396.
[3] P. Carpenter, C. Jin, D. Wilson, R. Zhang, D. Bohus, and A. Rudnicky, "Is this conversation on track," in *Eur. Conf. Speech Technology*, Aalborg, Denmark, Sep. 2001, pp. 2121–2124.
[4] F. Wessel, K. Macherey, and H. Ney, "A comparison of word graph and N-best list based confidence measures," *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 1587–1590, Jun. 2000.
[5] K. Hacioglu and W. Ward, "A concept graph based confidence measure," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Orlando, FL, May 2002, pp. 225–228.
[6] C. Neti, S. Roukos, and E. Eide, "Word-based confidence measures as a guide for stack search in speech recognition," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Munich, Germany, Apr. 1997, pp. 883–886.
[7] R. Sarikaya, H. Erdogan, Y. Gao, and M. Picheny, "Turn based language modeling for spoken dialog systems," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Orlando, FL, May 2002.
[8] R. Zhang and A. Rudnicky, "Word level confidence annotation using combination of features," in *Eur. Conf. Speech Technology*, Aalborg, Denmark, Sep. 2001.
[9] Y. Gao, H. Erdogan, Y. Li, V. Goel, and M. Picheny, "Recent advances in speech recognition system for IBM darpa communicator," in *Eur. Conf. Speech Technology*, Aalborg, Denmark, Sep. 2001.
[10] K. Davies *et al.*, "The IBM conversational telephony system for financial applications," in *Eur. Conf. Speech Technology*, Budapest, Hungary, Sep. 1999.
[11] F. Jelinek, J. Lafferty, D. Magerman, R. Mercer, A. Ratnaparkhi, and S. Roukus, "Decision tree parsing using a hidden derivational model," in *Proc. ARPA Human Language Technology Workshop*, 1994, pp. 272–277.
[12] S. Pradhan and W. Ward, "Estimating semantic confidence for spoken dialog systems," in *IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Orlando, FL, May 2002.
[13] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: Lattice based word error minimization," in *Eur. Conf. Speech Technology*, Budapest, Hungary, Sept. 1999, pp. 495–498.
[14] C. Pao, P. Schmid, and J. Glass, "Confidence scoring for speech understanding systems," in *Int. Conf. Spoken Language Processing*, Sydney, NSW, Australia, Dec. 1998.
[15] B. Maison and R. Gopinath, "Robust confidence annotation and rejection for continuous speech recognition," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 2001.
[16] T. Hazen, T. Burianek, J. Polifroni, and S. Seneff, "Recognition confidence scoring for use in speech understanding systems," in *Proc. Automatic Speech Recognition Workshop*, Paris, France, 2000, pp. 213–220.
[17] S. F. Chen and R. Rosenfeld, "A survey of smoothing techniques for maximum entropy models," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 1, pp. 37–50, Jan. 2000.
[18] S. Cox and S. Dasmahapatra, "High-level semantic approaches to confidence estimation in speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 10, no. 7, pp. 460–471, 2002.
[19] H. Erdogan, R. Sarikaya, Y. Gao, and M. Picheny, "Semantic structured language models for spoken dialog systems," *Comput. Speech Lang.*, to be published.
[20] C. Chelba and F. Jelinek, "Structured Language Modeling," *Comput. Speech Lang.*, vol. 14, no. 4, pp. 283–332, Oct. 2000.
[21] L. Chase, "Error-responsive feedback mechanisms for speech recognition," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, Apr. 1997.
[22] R. Rosenfeld, "Adaptive statistical language modeling: a maximum entropy approach," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, Apr. 1994.
[23] D. M. Magerman, "Natural Language Parsing as Statistical Pattern Recognition," Ph.D. dissertation, Stanford Univ., Stanford, CA, Feb. 1994.
[24] C. Popovici and P. Baggia, "Specialized language models using dialog predictions," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Munich, Germany, pp. 815–818.



Ruhi Sarikaya (M'01) received the B.S. degree from Bilkent University, Ankara, Turkey, in June 1995, the M.S. degree from Clemson University, Clemson, SC, in August 1997, and the Ph.D. degree from Duke University, Durham, NC in March 2001 all in electrical and computer engineering.

Since April 2001, he is a Research Staff Member in the Human Language Technologies Group at IBM T.J. Watson Research Center, Yorktown Heights, NY. From 1999 to 2001, he was a Researcher at the Center for Spoken Language Research at the University of Colorado at Boulder. In the summer of 1999, he worked at the Panasonic Speech Technology Laboratory, Santa Barbara, CA. His past and present research interests span speech recognition/enhancement, speech to speech translation, speaker identification/verification, natural language processing, and digital signal processing.

Dr. Sarikaya is a member of ACL and ISCA.

Yuqing Gao received the Ph.D. degree in electrical engineering from Southeastern University, Nanjing, China, in 1989.

From 1988 to 1989, she was a Visiting Scholar at National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing. From 1989 to 1992, she was a Research Staff Member of National Laboratory of Pattern Recognition as an Associate Professor (January 1991–February 1992) and an Assistant Professor (June 1989–January 1991), where she was the project leader and the chief scientist for several projects funded by NSF of China. From March 1992 until May 1993, she was a postdoctoral researcher at CRIN (Centre de Recherché en Informatique de Nancy), Nancy, France. From August 1993 to November 1995, she was a Research Staff Member and Project Manager for speech recognition research at Apple-ISS Research Center, Apple Computer, Inc., Singapore. Since 1995, she has been a Research Staff Member at IBM T. J. Watson Research Center, Yorktown Heights, NY, where she has been project leader for large vocabulary continuous speech dictation system and speech-to-speech translation research. He has published over 70 papers at various conferences and journals and contributed to four books. She holds 12 U.S. patents and has been principle investigator for DARPA funded research projects.

Dr. Gao has received several privilege awards from IBM, the Chinese Academy of Sciences, the State of Education Commission of China, and the State Council of China.

Michael Picheny (F'01) is the Manager of the Speech and Language Algorithms Group in the Human Language Technologies Group at the IBM T. J. Watson Research Center, Yorktown Heights, NY. He has worked in the speech recognition area since 1981. He has been heavily involved in the development of almost all of IBM's recognition systems, ranging from the world's first real-time large vocabulary discrete system through IBM's current ViaVoice product line. He has published numerous papers in both journals and conferences on almost all aspects of speech recognition. He is the co-holder of over 20 patents and was named a Master Inventor by IBM in 1995 and again in 2000.

Dr. Picheny served as an Associate Editor of the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING from 1986 to 1989 and is currently the Chairman of the Speech Technical Committee of the IEEE Signal Processing Society. He has received several awards from IBM for his work, including three Outstanding Technical Achievement Awards and two Research Division Awards.

Hakan Erdogan (M'99) received the B.S. degree in electrical engineering and mathematics in 1993 from METU, Ankara, Turkey. He received the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1995 and 1999, respectively. He worked on statistical tomographic image reconstruction techniques for his Ph.D.

He joined IBM T. J. Watson Research Center, Yorktown Heights, NY, in 1999. While there, he focused on statistical methods for acoustic and language modeling for speech recognition, dialog systems, and speech to speech translation. He joined Sabanci University, Istanbul, Turkey, in 2002. His research interests lie in the general area of statistical methods for multimedia information extraction and pattern classification with emphasis on audio processing.