Evaluating Discourse Understanding in Spoken Dialogue Systems

Ryuichiro Higashinaka[†], Noboru Miyazaki[‡], Mikio Nakano[†], and Kiyoaki Aikawa[§]
NTT Communication Science Laboratories, NTT Corporation.

This paper describes a method for creating an evaluation measure for discourse understanding in spoken dialogue systems. No well-established measure has yet been proposed for evaluating discourse understanding, which has made it necessary to evaluate it only on the basis of the system's total performance. Such evaluations, however, are greatly influenced by task domains and dialogue strategies. To find a measure that enables good estimation of system performance only from discourse understanding results, we enumerated possible discourse-understanding-related metrics and calculated their correlation with the system's total performance through dialogue experiments.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Discourse*; I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems—*Natural Language Interfaces*; H.5.2 [**Information interfaces and presentation**]: User interfaces—*Voice I/O*

General Terms: Languages, Measurement, Performance

Additional Key Words and Phrases: Discourse understanding, evaluation measures, speech understanding, spoken dialogue systems

1. INTRODUCTION

Due to advances in speech recognition and speech synthesis technologies, spoken dialogue systems, which exchange information with human users, have been attracting a lot of attention [McTear 2002; Zue and Glass 2000]. Such systems are used in many applications, such as train timetable systems [Sturm et al. 1999; Lamel et al. 2000], airline travel planning systems [Rudnicky et al. 2000; Doran et al. 2001; Seneff 2002], and call routing systems [Gorin et al. 1997; Chu-Carroll and Carpenter 1999]. What is common among these systems is that they need to understand a user's request. This paper focuses on this user request understanding phase rather than entire dialogues, which include negotiation with users and explanations of database content.

Unlike simple speech understanding systems that understand a single user utterance and

[†] 3-1 Morinosato-Wakamiya, Atsugi, Kanagawa 243-0198, Japan. {rh,nakano}@atom.brl.ntt.co.jp
[‡] Currently with NTT Cyber Space Laboratories, NTT Corporation. 1-1 Hikari-no-Oka, Yokosuka-shi, Kanagawa 239-0847, Japan. miyazaki.noboru@lab.ntt.co.jp
[§] Currently with the School of Media Science, Tokyo University of Technology. 1404-1 Katakuracho, Hachioji, Tokyo 192-0982, Japan. aik@media.teu.ac.jp

This paper is a modified and augmented version of our earlier reports [Higashinaka et al. 2002; 2003].

respond to it without taking context into account, spoken dialogue systems need to deal with multiple exchanges of utterances in the context of dialogues. This process is achieved by updating a dialogue state each time a user utterance is made. A dialogue state is a collection of bits of information that the system internally stores. Included in that information are the understanding result of the user utterances up to a certain point of time as well as grounding information, the user utterance history, and the system utterance history.

There is no well-established measure for evaluating discourse understanding. As a result, it has been evaluated only on the basis of the system's total performance, such as task completion rate, task completion time and user satisfaction estimated by questionnaires. However, such evaluations are greatly influenced by the task domains and dialogue strategies that the systems employ. This fact makes it difficult to compare various systems' discourse understanding. A measure that appropriately evaluates specifically discourse understanding capability would be useful for further improvement of discourse understanding components.

In evaluating single utterance understanding, which does not include discourse understanding, the *concept error rate* (CER) or the *keyword error rate* (KER) has been widely used as an evaluation measure [Glass et al. 2000]. Using the CER of discourse understanding results is one possibility. However, it may not be appropriate for the evaluation of discourse understanding, because it is unclear whether the CER correlates closely with the system's performance. A measure should have high correlation with what it is measuring. Since we seek to maximize the system's performance by improving the discourse understanding capability, the measure has to have high correlation with the system's total performance. Our aim is to find such a measure.

As our approach, we enumerate possible discourse-understanding-related metrics and obtain their correlation against the system performance through dialogue experiments using human subjects. The experiments have to be performed in several task domains utilizing various strategies to create a measure that can commonly be used across different systems. We can use the metric that has the highest correlation as the evaluation measure. It is also possible that the combination of some metrics will lead to a higher correlation. For such cases, we apply regression methods to create a single measure using the metric candidates. The methodology used here is similar to the one used in PARADISE [Walker et al. 1997] in that the impacts of various features of dialogues are assesed based on their correlation with the system's total performance. Instead of assorted features of dialogues, we focus specifically on discourse-related features and use their correlation to find appropriate evaluation measures for discourse understanding.

The next section briefly looks at the discourse understanding process in spoken dialogue systems. Section 3 explains the need for an evaluation measure for discourse understanding. In Section 4, using an example dialogue, we describe why conventional metrics, such as CER, cannot be used for the evaluation. In Section 5, our approach and various metrics concerning discourse understanding are described in detail. Then, in Section 6, we describe the dialogue experiments we performed to collect dialogue data using our dialogue systems. In Section 7, we show the correlation between each metric and the system performance and describe our attempt to create better measures using regression methods followed by detailed analysis of the obtained models. The paper concludes with a short summary and some recommendations.

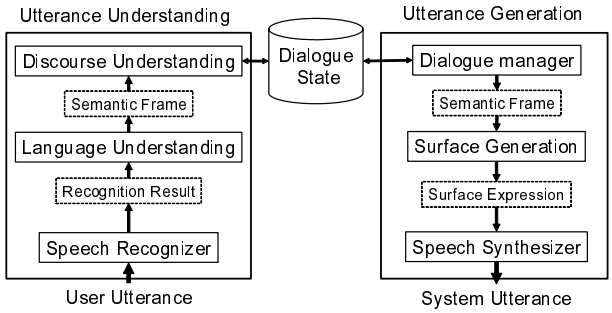

Fig. 1. Architecture of a spoken dialogue system.

2. DISCOURSE UNDERSTANDING IN SPOKEN DIALOGUE SYSTEMS

Here, we describe the basic architecture of a spoken dialogue system (Fig. 1). When receiving a user utterance, the system behaves as follows.

1. The speech recognizer receives a user utterance and outputs a speech recognition result, such as an N-best list and a word graph.
2. The language understanding component receives the speech recognition result. Syntactic and semantic analyses are performed to convert it into a meaning representation, often called a semantic frame, or, sometimes a logical form. A semantic frame is typically composed of a *dialogue act* that identifies the main intent of the user's utterance, augmented with necessary ancillary information, often encoded as attribute-value pairs, or using a predicate calculus terminology in the case of logical forms.
3. The discourse understanding component receives the semantic frame, refers to the current dialogue state, and updates the dialogue state.
4. The dialogue manager refers to the updated dialogue state, decides the next utterance, and outputs the next content to be delivered as a semantic frame. The dialogue state is updated at the same time so that it contains the content of system utterances.
5. The surface generation component receives the semantic frame and produces the surface expression (words).
6. The speech synthesizer receives the next words to be spoken and responds to the user by speech.

This paper focuses on the discourse understanding component. This component has to appropriately update the dialogue state so that the system can make as appropriate a response as possible. In this paper, we assume that a dialogue state can be expressed simply by a frame expression [Bobrow et al. 1977], which is common in many systems, also sometimes referred to as an electronic form or *E-form* [Goddeau et al. 1996]. A frame/E-form is a bundle of slots that consist of attribute-value pairs concerning a certain domain.

Figure 2 shows how the frames are updated in the course of a dialogue in a weather information system. In this example, the system has a frame consisting of three slots each representing place, date, and information type (general weather, temperature, and precipitation) respectively. At first, slot values are vacant. After several exchanges of utterances,

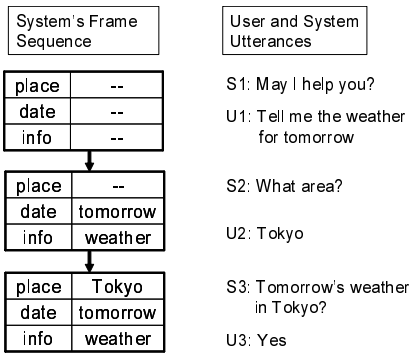

Fig. 2. An example of frame updates in a dialogue. (S means a system utterance and U a user utterance.)

the slots are updated and the system finally recognizes the correct user intention. Through the interactive process with the user, the frames get closer to the correct frame. Although recent dialogue management work [Komatani and Kawahara 2000; Dohaka et al. 2003] makes use of confidence scores to represent the correctness of frame and slot values, we only consider cases where the values can be either correct or incorrect in this paper. The update is performed based on discourse understanding rules that the discourse understanding component internally holds. Note that discourse understanding does not mean just filling slots using keywords contained in user utterances. For example, when the user denies some values in slots, the discourse understanding component might prepare alternative values, so that the dialogue manager can use them for suggestions. This can be achieved by looking back at the past exchanges of utterances stored in the dialogue state.

To represent dialogue states, plans have often been used [Allen and Perrault 1980; Carberry 1990]. Traditionally, plan-based discourse understanding methods have been implemented mostly in keyboard-based dialogue systems, although there have been some recent attempts to apply them to spoken dialogue systems [Allen et al. 2001; Rich et al. 2001]. However, considering the current performance of speech recognizers and the limitations in task domains, we believe frame-based discourse understanding and dialogue management are sufficient [Chu-Carroll 2000; Seneff 2002; Bobrow et al. 1977].

There are also object-oriented approaches for the modeling of dialogue states [Sparks et al. 1994; Abella and Gorin 1999]. Such approaches model dialogue states as objects that encapsulate the necessary information and behavior for achieving sub-tasks in a dialogue. The dialogue progresses by making transitions among the dialogue states until the task as a whole is complete. As long as the discourse understanding results can be represented by frames, for example, by aggregating the objects' information, we believe our approach can be applied to these models as well.

3. THE NEED FOR AN EVALUATION MEASURE IN DISCOURSE UNDERSTANDING

A qualitative measure for evaluating each component in spoken dialogue systems would be useful for improving components. Speech recognition and language understanding modules have been evaluated using the word error rate (WER) and the CER or KER, respectively. Although they may not be the best evaluation measures for those components, they

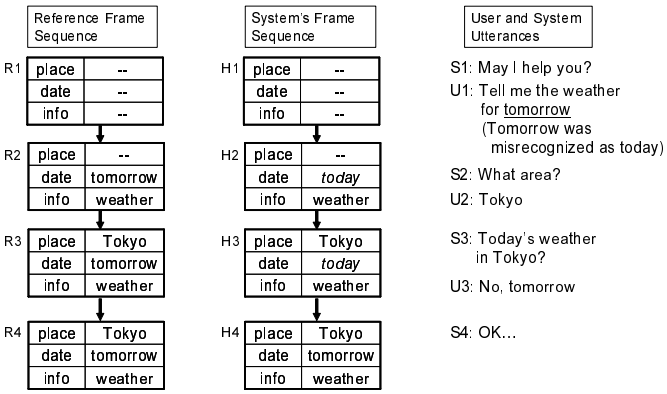

Fig. 3. An example of frame updates with corresponding reference frames. (R and H mean a reference frame and a hypothesis frame respectively.)

are intuitively reasonable and widely used among researchers and developers. Thanks to these measures, we can compare the speech recognition and language understanding components of various systems dealing with various tasks and strategies.

However, no well-established measure for discourse understanding exists, which makes it necessary to evaluate discourse understanding on the basis of the system's total performance, such as task completion rate, task completion time, and user satisfaction. Since the system's total performance varies depending on the system's task domains and dialogue strategies, when we compare several different discourse understanding components, task domains and dialogue strategies have to be fixed. Experiments have to be redone whenever the settings are changed, which makes the evaluation cost high.

A measure that can appropriately and specifically evaluate the discourse understanding capability would be useful for reducing the cost of dialogue experiments and making clear the performance of discourse understanding of various systems. Therefore, there is a strong need for an evaluation measure for discourse understanding.

4. PROBLEM

What is an appropriate measure for discourse understanding? One candidate is the CER of a system's frames. However, this measure may not be suitable because its degree of correlation with system performance is uncertain. Since we seek to maximize the system's performance by improving discourse understanding, the measure should correlate highly with the system's total performance. There may be other measures that have higher correlation, and the straightforward use of the CER may lead to inappropriate evaluation.

Figure 3 shows an illustrative example of the problem. We call a system's frame a *hypothesis frame* and the correct frame that can be annotated later a *reference frame*. As a reference frame, we use the ideal discourse understanding result that takes all previous system and user utterances into account instead of using a frame that can be reached from the previous hypothesis frame and the succeeding user utterance.

In the example, part of the user's second utterance "tomorrow" is misrecognized as "today", and the system updated the initial frame (H1) to an incorrect frame (H2). Even

after the user's next utterance "Tokyo", the wrong value "today" is still in the date slot (H3). This misunderstood item is later corrected by the user, who notices the error in the frame because of the incorrect system confirmation and corrects the value (H4). R1 to R4 show the references for the corresponding hypothesis frames.

We want to evaluate the system's frame sequence, which results from discourse understanding. For systems that do not handle previous utterances, the CER is suitable for evaluating utterance understanding because the situation is similar to evaluating a single utterance understanding. However, when we take previous frames into account, the suitability of the CER becomes unclear. For example, there are cases where the resulting frame is wrong, but it may have been updated correctly in part.

Consider three metrics: the slot error rate, the update precision, and the CER. The slot error rate is the rate of wrong values in a frame. The update precision shows the ratio of incorrect slots within updated slots. The CER is the ratio of incorrect slots over the number of filled slots. Our definition of the CER may seem different from the one commonly used. However, since we focus on the discourse understanding result (a frame) instead of attribute-value pairs contained in previous user utterances, the number of substituted slots, deleted slots, and inserted slots over the number of filled slots can be considered to match the definition of the CER. They are derived as shown below.

(1) Slot error rate

$$\frac{\#\text{ of incorrect slots}}{\#\text{ of slots}}$$

(2) Update precision

$$\frac{\#\text{ of correctly updated slots}}{\#\text{ of updated slots}}$$

(3) CER

$$\frac{\#\text{ of incorrect slots}}{\#\text{ of filled slots}}$$

The value of each metric is calculated for each pair of hypothesis and reference frames. In the example (Fig. 4), the average slot error rate for the hypothesis frames is $(1/3 + 1/3 + 0/3)/3 = 0.22$, the update precision is $(1/2 + 1/1 + 1/1)/3 = 0.83$, and the CER is $(1/2 + 1/3 + 0/3)/3 = 0.28$. These values encode the discourse understanding in some way. The slot error rate seems a reasonable measure, because a frame is the final result from the discourse understanding component and forms the basis for the next system utterance. However, notice that the slot that has an erroneous value "today" is inherited and counted as an error twice. The update precision, on the other hand, focuses only on the updated slots, avoiding the shortcomings of the slot error rate, but the entire frame is not taken into consideration. The CER is only different from the slot error rate in that it focuses on the filled slots; it has the same double counting problem.

Currently, it is not clear whether the evaluation should focus on the frames themselves or the way they are updated in a dialogue, which makes it difficult to decide the most reasonable metric. Moreover, there may be other metrics that are more appropriate for the evaluation.

5. APPROACH

As our approach, we enumerate possible metrics concerning frame sequences and choose those that have good correlation with the system's performance as evaluation measures. We also combine the enumerated metrics to create a single measure by regression methods. It is likely that such a combined measure will have higher correlation by taking many aspects of frame values and updates into account.

Then, we perform dialogue experiments using human subjects and obtain both the value of each metric and the total performance of a dialogue. After collecting sufficient dialogue data, we calculate the correlation between the value of each metric and the total performance of dialogues. We treat the metric that has the highest correlation as the appropriate evaluation measure. In addition, by combining the metric values, we create a single measure by regression methods using all the values of metric candidates as explaining variables and the total performance as the explained variable.

To find a measure that is commonly applicable independent of task domains and dialogue strategies, dialogue experiments have to be performed using different task domains and different dialogue strategies.

5.1 Metric Candidates

Besides the slot error rate, update precision, and the CER, we came up with additional metrics that can be categorized into five groups depending on the viewpoints. They are all calculable by comparing hypothesis frames and reference frames. We consider it necessary that the evaluation can be achieved by simple calculation, such as by comparing the hypothesis frame with the reference frame, so that it can be easily applied to various systems by developers and researchers in the field. In this paper, the value of each metric in a dialogue is represented by the average value.

1. *Metrics concerning slot values:* Metrics comparing the values of every slot of a hypothesis frame with that of a reference frame. The slot error rate is one of them. And slot accuracy, insertion error rate, deletion error rate, and substitution error rate also are considered.
2. *Metrics concerning updated slot values:* Metrics comparing the values of only the updated slots. With these metrics, we can avoid the double counting of inherited errors in slots. Update precision is one of them. The calculation is performed in two ways. One concerns the correctness of updated slots in a hypothesis frame, the other the correctness of those in the reference frame. For example, update precision is the ratio of correctly updated slots in the updated slots, whereas update recall is the ratio of correctly updated slots over the slots that should be updated.
3. *Metrics concerning filled slot values:* Sometimes the aim of a task is not to fill every slot but to fill some of them. To reflect such cases, we propose metrics that focus only on the filled slots. These metrics are calculated for a hypothesis frame and a reference frame.
4. *CER:* The same as the conventional CER. It expresses the correctness of filled slots. The difference from the metrics concerning the filled slots is that this metric includes the insertion error. This metric is also calculated for a hypothesis frame and a reference frame.
5. *A metric concerning a frame sequence:* If the user intention is exactly recognized by

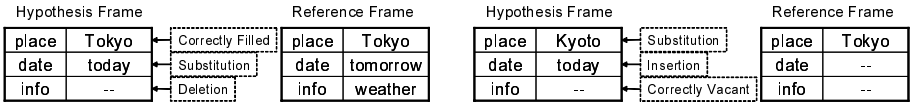

Fig. 4. *Labeling the slot values of a hypothesis frame.*

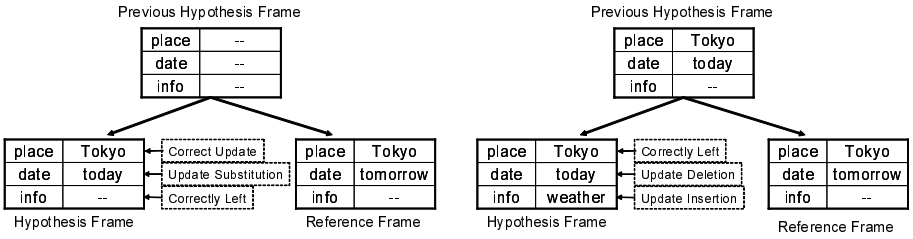

Fig. 5. *Labeling the update of a hypothesis frame.*

Table I. *Labels given to each slot of a hypothesis frame.*

Correctly Vacant	$Ref = Hyp$ and $Ref = Null$
Correctly Filled	$Ref = Hyp$ and $Ref \neq Null$
Insertion	$Ref = Null$ and $Hyp \neq Null$
Deletion	$Ref \neq Null$ and $Hyp = Null$
Substitution	$Ref \neq Hyp$ and $Ref \neq Null$ and $Hyp \neq Null$

the system, dialogue management is likely to work effectively, leading to improvement of total performance. Therefore, we propose a frame match rate, which is the rate that the hypothesis frame is exactly the same as the reference frame in all frames in a dialogue.

5.2 Labeling

Here, we describe the procedure for deriving the values of metric candidates explained in the previous section. First, we label each slot of the hypothesis frame by comparing the corresponding slot in the reference frame. The reference frame has to be hand-crafted in advance.

The comparison is performed in two ways. One is a simple comparison of each value of the slots performed to see if the values are the same or different or if the slots have values at all. From this comparison, each slot of a hypothesis frame is given one of five labels (Fig. 4). Table I shows the labeling scheme. In the table, values of a certain slot of a hypothesis frame, a reference frame, and the previous frame are written as Hyp, Ref, and $Prev$, respectively. If a slot does not have a value, it is denoted as $Null$.

The other comparison is performed for changes from the previous hypothesis frame; "the difference between the previous hypothesis frame and the current hypothesis frame" is compared with "the difference between the previous hypothesis frame and the reference frame". From this comparison, one of five labels is assigned to each slot of a hypothesis frame (Fig. 5). Table II shows the labeling scheme.

Table II. *Labels given to the update of each slot of a hypothesis frame.*

Correctly Left	$Prev = Ref$ and $Prev = Hyp$ and $Ref = Hyp$
Correct Update	$Prev \neq Ref$ and $Prev \neq Hyp$ and $Ref = Hyp$
Update Insertion	$Prev = Ref$ and $Prev \neq Hyp$
Update Deletion	$Prev \neq Ref$ and $Prev = Hyp$
Update Substitution	$Prev \neq Ref$ and $Prev \neq Hyp$ and $Ref \neq Hyp$

5.3 List of Metric Candidates

From the ten labels, we derive the values of metric candidates. The derivation formulae are presented below, where CV, CF, I, D, S, CU, CL, UI, UD, and US represent the number of slots labeled Correctly Vacant, Correctly Filled, Insertion, Deletion, Substitution, Correct Update, Correctly Left, Update Insertion, Update Deletion, and Update Substitution respectively. There are 26 metric candidates in all.

Metrics concerning slot values:

1. Slot accuracy
$$\frac{CV + CF}{CV + CF + I + D + S}$$
2. Insertion error rate
$$\frac{I}{CV + CF + I + D + S}$$
3. Deletion error rate
$$\frac{D}{CV + CF + I + D + S}$$
4. Substitution error rate
$$\frac{S}{CV + CF + I + D + S}$$
5. Slot error rate
$$\frac{I + D + S}{CV + CF + I + D + S}$$

Metrics concerning updated slot values in a hypothesis frame:

6. Update precision
$$\frac{CU}{CU + US + UI}$$
7. Correctly remaining rate in hypothesis
$$\frac{CL}{CL + UD}$$
8. Update insertion error rate in hypothesis
$$\frac{UI}{CU + US + UI}$$
9. Update deletion error rate in hypothesis
$$\frac{UD}{CL + UD}$$
10. Update substitution error rate in hypothesis
$$\frac{US}{CU + US + UI}$$

Metrics concerning updated slot values in a reference frame:

11. Update recall
$$\frac{CU}{CU + US + UD}$$
12. Correctly remaining rate in reference
$$\frac{CL}{CL + UI}$$
13. Update insertion error rate in reference
$$\frac{UI}{CL + UI}$$
14. Update deletion error rate in reference
$$\frac{UD}{CU + US + UD}$$
15. Update substitution error rate in reference
$$\frac{US}{CU + US + UD}$$

Metrics concerning filled slot values in a hypothesis frame:

16. Slot accuracy for filled slots in hypothesis
$$\frac{CF}{CF + I + S}$$
17. Insertion error rate for filled slots in hypothesis
$$\frac{I}{CF + I + S}$$

18. Substitution error rate for filled slots in hypothesis
$$\frac{S}{CF+I+S}$$
19. Slot error rate for filled slots in hypothesis
$$\frac{I+S}{CF+I+S}$$

Metrics concerning filled slot values in a reference frame:

20. Slot accuracy for filled slots in reference
$$\frac{CF}{CF+D+S}$$
21. Deletion error rate for filled slots in reference
$$\frac{D}{CF+D+S}$$
22. Substitution error rate for filled slots in reference
$$\frac{S}{CF+D+S}$$
23. Slot error rate for filled slots in reference
$$\frac{D+S}{CF+D+S}$$

CER:

24. CER for a hypothesis frame
$$\frac{I+D+S}{CF+I+S}$$
25. CER for a reference frame
$$\frac{I+D+S}{CF+D+S}$$

A metric concerning a frame sequence:

26. Frame match rate
$$\frac{\#\text{ of exactly correct frames}}{\#\text{ of frames}}$$

5.4 Performance Measure

System performance has been evaluated in many ways, but it is not certain what really is a valid system performance measure. We chose task completion time and user satisfaction, whose values are commonly used for system evaluations.

In this research, the aim of a dialogue is to complete a task. Efficiently completing a task is an important factor in improving system performance. Therefore, we employ task completion time to represent the performance of a dialogue. We also employ user satisfaction as determined by questionnaires, a method used by many researchers, such as [Walker et al. 2000]. Although there is always controversy concerning the validity of questionnaires to estimate user satisfaction, no alternatives have been proposed.

6. DATA COLLECTION

6.1 Systems

We created three systems to perform the dialogue experiments for data collection. One is in a weather information service domain (**WI**), and the other two are in a meeting room reservation domain (**MR-1**, **MR-2**).

WI provides Japan-wide weather information. Users specify a prefecture name, a city name, a date, and an information type (general weather, temperature, and precipitation) to obtain the desired information. The system has four slots for understanding. It has a speech recognition vocabulary of 853. The language model is a trigram trained from the randomly generated texts of acceptable phrases.

MR-1 and MR-2 provide meeting room reservation service. Users specify a date, a room, and start and end times for the reservation. The systems has four slots for understanding. Both have a speech recognition vocabulary of 243. The language model is a trigram trained from the transcription obtained in advance using the same system. The difference between MR-1 and MR-2 lies in their discourse understanding components. Both

systems create multiple dialogue state candidates ordered by priority after each user utterance and choose the highest ranked one as the best dialogue state. When deciding the best dialogue state, MR-1 preserves lower ranked dialogue states, whereas MR-2 discards them totally (See [Higashinaka et al. 2003] for details).

All three systems were developed using the spoken dialogue system toolkit WIT [Nakano et al. 2000]. Their speech recognition engine is Julius [Lee et al. 2001] used with its attached acoustic model, and the speech synthesis engine is FinalFluet [Takano et al. 2001]. Each system has two switchable dialogue strategies. One is to keep accepting user utterances until it has enough information to fulfill a task or the user explicitly requests a system response. The other is to confirm each user utterance.

6.2 Experiment

Using the three systems, we collected dialogue data for analysis. The dialogue data were collected using naive users in acoustically insulated booths.

Twelve subjects used WI. Each subject was given a task sheet listing what should be requested. They were instructed to complete the tasks one by one. We prepared eight task patterns. Together with the two dialogue strategies, each subject performed 16 dialogues, for a total 192 dialogues collected. Twenty-eight subjects used MR-1 and MR-2. Using four task patterns, two dialogue strategies, and two systems, each performed 16 dialogues, and 448 dialogues were collected.

After completing each dialogue, each subject was asked to fill out a questionnaire; the same one used in [Walker et al. 2000]. The questionnaire is composed of nine questions concerning text to speech (TTS) performance, automatic speech recognition (ASR) performance, task ease, interaction pace, user expertise, system response, expected behavior, comparable interface, and future use and is on a 1-to-7 Likert scale.

We recorded system utterances, start and end times of user utterances, and dialogue states before and after the user utterance. The user's voice and system's voice were also recorded, and all user utterances were transcribed. Dialogues in which it took more than three minutes to complete the task were treated as failures. Task completion rates for WI, MR-1 and MR-2 were 95.8% (185/192), 91.1% (204/224), and 88.4% (198/224), respectively. The word error rates (WER) for WR and MR-1+MR-2 were 30.01% and 33.92%, respectively.

We hand-annotated reference frames. To avoid a large hand-labeling effort, we prepared an annotating tool that processes transcriptions to generate pre-reference frames, which were later corrected by human labellers. The correction took several hours for our dialogue data. Then, using the labeling scheme, we labeled each slot of the corresponding hypothesis frame and obtained all 26 values of the metrics for each dialogue. Task completion times were normalized using task patterns and dialogue strategies because task completion time can be greatly influenced by them. We used the total score of the questionnaire to represent user satisfaction.

7. DATA ANALYSIS

7.1 Correlations of the Metric Candidates

Table III shows the correlation coefficients of the 26 metrics against task completion time and user satisfaction. These are the results obtained when we used all the data: WI, MR-1, and MR-2. (Hereafter, we use WI + MR-1 + MR-2 to express the combined data of

the systems.) For analysis, we used only successful dialogues for which task completion times and user satisfaction data were available. In WI + MR-1 + MR-2, there were 584 samples for task completion time after removing three 3-σ outliers, and 587 samples for user satisfaction.

The update recall has relatively high correlation with a correlation coefficient -0.647 followed by -0.607 of frame match rate and -0.579 of update precision. The tendency is similar for user satisfaction, although the correlation coefficients are basically lower. By simple linear regression analysis with ten-fold cross validation, we found that update precision, update recall, and frame match rate explain 32.8%, 41.32%, and 36.31% of task completion time and 11.89%, 18.81% and 15.73% of user satisfaction respectively.

As a result, we can say that the update recall, frame match rate, and update precision are strong candidates for evaluation measures, especially the update recall.

Table III. Correlation coefficients (R) of the 26 metrics against task completion time and user satisfaction.

		Task completion time	User satisfaction
1.	Slot accuracy	-0.554	0.336
2.	Insertion error rate	0.117	-0.004
3.	Deletion error rate	0.318	-0.210
4.	Substitution error rate	0.450	-0.294
5.	Slot error rate	0.554	-0.336
6.	Update precision	**-0.579**	**0.358**
7.	Correctly remaining rate in hypothesis	-0.437	0.309
8.	Update insertion error rate in hypothesis	0.326	-0.179
9.	Update deletion error rate in hypothesis	0.437	-0.309
10.	Update substitution error rate in hypothesis	0.451	-0.296
11.	Update recall	**-0.647**	**0.441**
12.	Correctly remaining rate in reference	-0.182	0.099
13.	Update insertion error rate in reference	0.572	-0.397
14.	Update deletion error rate in reference	0.182	-0.099
15.	Update substitution error rate in reference	0.386	-0.247
16.	Slot accuracy for filled slots in hypothesis	-0.458	0.237
17.	Insertion error rate for filled slots in hypothesis	0.127	-0.007
18.	Substitution error rate for filled slots in hypothesis	0.441	-0.271
19.	Slot error rate for filled slots in hypothesis	0.458	-0.237
20.	Slot accuracy for filled slots in reference	-0.542	0.330
21.	Deletion error rate for filled slots in reference	0.308	-0.199
22.	Substitution error rate for filled slots in reference	0.452	-0.263
23.	Slot error rate for filled slots in reference	0.542	-0.330
24.	CER for a hypothesis frame	0.430	-0.247
25.	CER for a reference frame	0.404	-0.214
26.	Frame match rate	**-0.607**	**0.406**

7.2 Obtained Regression Models

We used two regression methods to create a single evaluation measure: *multiple linear regression (MLR)* and *support vector regression (SVR)*. For the MLR, the $m5'$ method [Wang and Witten 1997; Witten and Frank 1999] was used for attribute selection instead of the greedy method. SVR is an optimization-based approach for solving machine learning re-

Table IV. *Squared correlation coefficients (R^2) and the root mean square error (RMSE) (in brackets) for multiple linear regression (MLR) and support vector regression (SVR).*

	Task completion time		User satisfaction	
	MLR	SVR	MLR	SVR
WI	0.444 (0.718)	0.318 (0.809)	0.125 (1.514)	0.172 (1.470)
MR-1	0.409 (0.704)	0.445 (0.688)	0.158 (1.220)	0.212 (1.168)
MR-2	0.510 (0.690)	0.506 (0.708)	0.247 (1.114)	0.255 (1.105)
MR-1 + MR-2	0.474 (0.684)	0.483 (0.690)	0.198 (1.167)	0.245 (1.128)
WI + MR-1 + MR-2	0.429 (0.717)	**0.440** (0.724)	0.180 (1.284)	**0.195** (1.275)

gression problems based on support vector machines [Vapnik 1995; Smola and Schölkopf 1998; Chang and Lin 2001]. We used a polynomial kernel expressed as

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d \text{ where } d = 2 \quad (1)$$

We created regression models for each regression method using *task completion time normalized by the task pattern and the dialogue strategy*, and *user satisfaction* as the explained variables and the 26 metrics as explaining variables.

Table IV shows squared correlation coefficients (R^2) and the root mean square error ($RMSE$) for the two regression methods. These are the results of ten-fold cross validation. When task completion time is the explained variable, most of the obtained regression models fit comparatively well and show validity as evaluation measures. For user satisfaction, the fit is not as good. The performance of SVR is similar to that of MLR.

One may notice that the regression models for MR-1+MR-2 perform better on both task completion time and user satisfaction than for WI. This is because, in WI, for certain city names, repeated misrecognition happened, which caused the system to have slots that have been substituted in the same dialogue a number of times. Since the metric values are represented by their average values in a dialogue, there are often cases where the ratio of substitution error is the same, but the number of times the error occured is not. In such cases, it is difficult for regression models to achieve high correlation. As evidence, when we examined the indivisual correlation of each metric with system performance for WI and MR-1+MR-2, we found that the slot substitution error rate accounts for 11.3% of user satisfaction for MR-1+MR-2, whereas it accounts for only 0.19% of user satisfaction for WI.

Figure 6 shows the distribution of actual and predicted task completion times for the acquired model using WI + MR-1 + MR-2. The grouping of data, which appears as a horizontal line just above -1.0 in the vertical axis, means that dialogues with different actual task completion times were forcefully mapped to the same task completion times by the regression model since they have identical discourse understanding characteristics. This is attributable to possible differences in the duration of user pauses and speech intervals among the subjects and the limitations of using the average values of the metrics in a dialogue as the discourse features.

In the case of WI + MR-1 + MR-2, the obtained regression models explain 44 % of the task completion time, and 19.5 % of user satisfaction. In comparison with the case of a single metric, the regression methods provide a slightly better prediction of system performance.

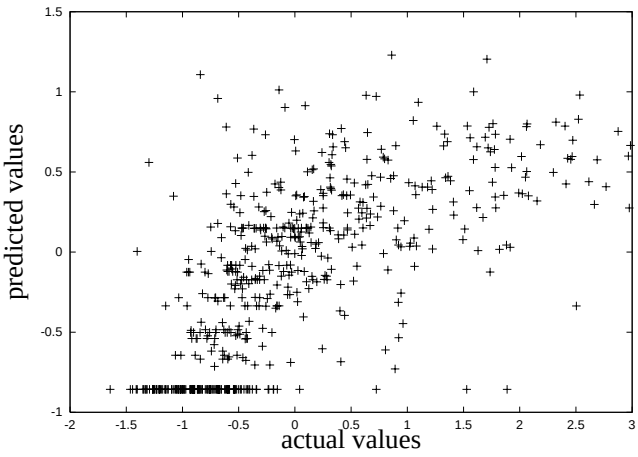

Fig. 6. *Distribution of actual and predicted task completion times by the support vector regression (SVR) model trained from WI + MR-1 + MR-2.*

Table V. *Commonality between the trained support vector regression (SVR) models for task completion time. (Squared correlation coefficients (*R^2*) and the root mean square error (*RMSE*) in brackets.)*

Training data \ Test data	WI	MR-1	MR-2	MR-1 + MR-2	WI + MR-1 + MR-2
WI	–	0.268 (0.794)	0.369 (0.792)	0.320 (0.793)	0.387 (0.749)
MR-1	0.342 (1.050)	–	0.368 (0.812)	0.436 (0.718)	0.366 (0.837)
MR-2	0.084 (1.349)	0.350 (0.789)	–	0.474 (0.702)	0.257 (0.956)
MR-1 + MR-2	0.262 (0.999)	0.504 (0.636)	0.568 (0.647)	–	0.391 (0.773)
WI + MR-1 + MR-2	0.495 (0.681)	0.487 (0.648)	0.523 (0.678)	0.503 (0.663)	–

7.3 Commonality in Regression Models

To check whether a regression model trained from the data of one domain/system has commonality with that of another, we calculated R^2 and $RMSE$ for every combination of models. Table V shows the results for the SVR models with task completion time as the explained variable. Most of the R^2 values are around 0.4, suggesting that the model of one domain can be safely applied to that of the other. Since the performance of the model trained from WI + MR-1 + MR-2 shows sufficient performance against other models, this model can be used as a reasonable discourse evaluation measure. For this reason, hereafter, we only deal with models trained from WI + MR-1 + MR-2. Table VI shows the results for SVR models when user satisfaction is used as the explained variable. The tendency is similar for the MLR models.

7.4 Important Factor Analysis of Regression Models

Analyzing the obtained SVR models allows us to list up the possible major metrics for the prediction of the explained variables [Hirao et al. 2002]. First, the objective function of SVR is defined as

Table VI. *Commonality between the trained support vector regression (SVR) models for user satisfaction. (Squared correlation coefficients (R^2) and the root mean square error (RMSE) in brackets.)*

Training data \ Test data	WI	MR-1	MR-2	MR-1 + MR-2	WI + MR-1 + MR-2
WI	–	0.183 (2.407)	0.177 (2.438)	0.185 (2.414)	0.211 (2.122)
MR-1	0.245 (2.502)	–	0.247 (1.320)	0.249 (1.309)	0.254 (1.395)
MR-2	0.247 (2.347)	0.262 (1.202)	–	0.277 (1.176)	0.283 (1.236)
MR-1+MR-2	0.246 (2.426)	0.253 (1.258)	0.262 (1.250)	–	0.267 (1.317)
WI+MR-1+MR-2	0.193 (2.299)	0.187 (1.649)	0.195 (1.625)	0.199 (1.612)	–

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i;\mathbf{x}_i \in SVs} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \\ &= \sum_{i;\mathbf{x}_i \in SVs} \alpha_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) + b \\ &= \mathbf{w} \cdot \phi(\mathbf{x}) + b \end{aligned} \tag{2}$$

where SVs is the set of support vectors, and $\phi(\mathbf{x})$ an explicit representation of new feature vectors $\mathbf{x}$ mapped in the new feature space by the kernel. In the case of the 26 dimensions (features) in our original space and using a second-degree polynomial kernel, the dimensions of the new feature space become 378, and $\mathbf{w}$ is written as

$$\begin{aligned} \mathbf{w} = \big(& \sum_{i;\mathbf{x}_i \in SVs} \alpha_i x_{i1}^2, \ldots, \sum_{i;\mathbf{x}_i \in SVs} \alpha_i x_{i26}^2, \\ & \sqrt{2} \sum_{i;\mathbf{x}_i \in SVs} \alpha_i x_{i1} x_{i2}, \ldots, \sqrt{2} \sum_{i;\mathbf{x}_i \in SVs} \alpha_i x_{i25} x_{i26}, \\ & \sqrt{2} \sum_{i;\mathbf{x}_i \in SVs} \alpha_i x_{i1}, \ldots, \sqrt{2} \sum_{i;\mathbf{x}_i \in SVs} \alpha_i x_{i26}, 1 \big) \end{aligned} \tag{3}$$

where $x_{i1} \cdots x_{i26}$ are the values of the 26 metrics of the ith support vector. By gathering up the weighting factors by the metrics and by the combination of the metrics, we obtain the following weights:

$$\begin{aligned} W(x_1) &= \sum_{i;\mathbf{x}_i \in SVs} \alpha_i x_{i1}^2 + \sqrt{2} \sum_{i;\mathbf{x}_i \in SVs} \alpha_i x_{i1} \\ &\vdots \\ W(x_{26}) &= \sum_{i;\mathbf{x}_i \in SVs} \alpha_i x_{i26}^2 + \sqrt{2} \sum_{i;\mathbf{x}_i \in SVs} \alpha_i x_{i26} \\ W(x_1, x_2) &= \sqrt{2} \sum_{i;\mathbf{x}_i \in SVs} \alpha_i x_{i1} x_{i2} \\ &\vdots \\ W(x_{25}, x_{26}) &= \sqrt{2} \sum_{i;\mathbf{x}_i \in SVs} \alpha_i x_{i25} x_{i26} \end{aligned}$$

We calculated all 351 weights $({}_{26}C_2 + 26)$ from the obtained models. Tables VII and

Table VII. *Five dominating weighting factors for the prediction of task completion time.*

1.	W(*Update precision*)	-0.154
2.	W(*Update recall*)	-0.108
3.	W(*Update precision*,*Update recall*)	-0.092
4.	W(*Deletion error rate for updated slots in reference*)	0.084
5.	W(*Update precision*,*Slot accuracy for filled slots in reference*)	-0.080

Table VIII. *Five dominating weighting factors for the prediction of user satisfaction.*

1.	W(*Update recall*)	0.144
2.	W(*Update recall*,*Frame match rate*)	0.101
3.	W(*Update precision*,*Update recall*)	0.101
4.	W(*Frame match rate*)	0.100
5.	W(*Correctly remaining rate in hypothesis*,*Update recall*)	0.086

VIII show the five dominant metrics or combinations of metrics for each model with their weights. The higher the weights are, the more significant the metrics or the combinations of metrics become. From the tables, one can see that the update precision plays a key role in the prediction of task completion time, and the update recall is the most important factor for improving user satisfaction. The frame match rate is also important for user satisfaction.

8. CONCLUDING REMARKS

This paper presented a method for creating an evaluation measure for discourse understanding in spoken dialogue systems. We enumerated metric candidates for the evaluation of discourse understanding and calculated their correlation with the system's performance through dialogue experiments. We also created a single evaluation measure combining the metrics by regression methods to create a better measure. We found that update recall, frame match rate, and update precision had relatively good correlation with system performance, suggesting they are appropriate as evaluation measures. Above all, update recall can explain 41.3 % of the task completion time, and 18.8 % of user satisfaction. The use of the multiple linear regression (MLR) and support vector regression (SVR) methods revealed that the weighted sum of the metric values can create a measure that performs slightly better than a single metric. With the obtained regression model, 44 % of the task completion time and 19.5 % of user satisfaction can be explained. An analysis of the obtained SVR models also revealed that the update recall, update precision, and frame match rate play important roles in improving system performance.

Overall, we found that user satisfaction is more difficult to predict than task completion time. This can be attributable to the fact that we are dealing with task-oriented dialogues and that there exists a large variety of questionnaire results among subjects.

To conclude, we suggest using the update recall as an evaluation measure for discourse understanding in spoken dialogue systems. Update precision can also be used to support the evaluation. Considering that it is now common practice to combine the precision and recall metrics into an overall F-measure (harmonic mean) and that the two metrics are strong candidates for evaluation measures, the use of the F-measure can also be considered. In fact, we found that the F-measure has a higher correlation than the update precision and the update recall alone and explains 43.4% of the task completion time. Therefore, in cases where the two metrics are available, we recommend the F-measure be used. We do

not encourage the use of the obtained regression models as evaluation measures because they only offer a slight improvement and because we believe that the measure should be as simple as possible. With the measure, we can safely compare discourse understanding components of various spoken dialogue systems that deal with different task domains and dialogue strategies.

ACKNOWLEDGMENTS

We thank many people for their helpful comments. They include Shoji Makino, Kohji Dohsaka, Norihito Yasuda, Kentaro Ishizuka, Katsuhito Sudoh, and Matthias Denecke. We also thank Atsushi Fukayama and Jun Suzuki for advise on the multiple linear regression and support vector regression, respectively. Finally, we thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

ABELLA, A. AND GORIN, A. L. 1999. Construct algebra: Analytical dialogue management. In *Proc. 37th ACL*. 191–199.

ALLEN, J., FERGUSON, G., AND STENT, A. 2001. An architecture for more realistic conversational systems. In *Proc. IUI*. 1–8.

ALLEN, J. F. AND PERRAULT, C. R. 1980. Analyzing intention in utterances. *Artif. Intel. 15*, 143–178.

BOBROW, D. G., KAPLAN, R. M., KAY, M., NORMAN, D. A., THOMPSON, H., AND WINOGRAD, T. 1977. GUS, a frame driven dialog system. *Artif. Intel. 8*, 155–173.

CARBERRY, S. 1990. *Plan Recognition in Natural Language Dialogue*. MIT Press, Cambridge, Mass.

CHANG, C.-C. AND LIN, C.-J. 2001. *LIBSVM: a library for support vector machines*. Software available at *http://www.csie.ntu.edu.tw/˜cjlin/libsvm*.

CHU-CARROLL, J. 2000. MIMIC: An adaptive mixed initiative spoken dialogue system for information queries. In *Proc. 6th Applied NLP*. 97–104.

CHU-CARROLL, J. AND CARPENTER, B. 1999. Vector-based natural language call routing. *Comp. Ling. 25,* 3, 361–388.

DOHAKA, K., YASUDA, N., AND AIKAWA, K. 2003. Efficient spoken dialogue control depending on the speech recognition rate and system's database. In *Proc. Eurospeech*. 657–660.

DORAN, C., ABERDEEN, J., DAMIANOS, L., AND HIRSCHMAN, L. 2001. Comparing several aspects of human-computer and human-human dialogues. In *Proc. SIGDIAL*. 48–57.

GLASS, J., POLIFRONI, J., SENEFF, S., AND ZUE, V. 2000. Data collection and performance evaluation of spoken dialogue systems: The MIT experience. In *Proc. ICSLP*. 1–4.

GODDEAU, D., MENG, H., POLIFRONI, J., SENEFF, S., AND BUSAYAPONGCHAI, S. 1996. A form-based dialogue manager for spoken language applications. In *Proc. ICSLP*. 701–704.

GORIN, A. L., RICCARDI, G., AND WRIGHT, J. H. 1997. How may I help you? *Speech Comm. 23*, 113–127.

HIGASHINAKA, R., MIYAZAKI, N., NAKANO, M., AND AIKAWA, K. 2002. A method for evaluating incremental utterance understanding in spoken dialogue systems. In *Proc. ICSLP*. 829–832.

HIGASHINAKA, R., MIYAZAKI, N., NAKANO, M., AND AIKAWA, K. 2003. Evaluating discourse understanding in spoken dialogue systems. In *Proc. Eurospeech*. 1941–1944.

HIGASHINAKA, R., NAKANO, M., AND AIKAWA, K. 2003. Corpus-based discourse understanding in spoken dialogue systems. In *Proc. 41st ACL*. 240–247.

HIRAO, T., ISOZAKI, H., MAEDA, E., AND MATSUMOTO, Y. 2002. Extracting important sentences with support vector machines. In *Proc. 19th COLING*. 342–348.

KOMATANI, K. AND KAWAHARA, T. 2000. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proc. 18th COLING*. Vol. 1. 467–473.

LAMEL, L., ROSSET, S., GAUVAIN, J., BENNACEF, S., GARNIER-RIZET, M., AND PROUTS, B. 2000. The LIMSI ARISE system. *Speech Comm. 31*, 339–353.

LEE, A., KAWAHARA, T., AND SHIKANO, K. 2001. Julius – an open source real-time large vocabulary recognition engine. In *Proc. Eurospeech*. 1691–1694.

MCTEAR, M. 2002. Spoken dialogue technology: enabling the conversational interface. *ACM Computing Surveys 34*, 90–169.

NAKANO, M., MIYAZAKI, N., YASUDA, N., SUGIYAMA, A., HIRASAWA, J., DOHSAKA, K., AND AIKAWA, K. 2000. WIT: A toolkit for building robust and real-time spoken dialogue systems. In *Proc. SIGDIAL*. 150–159.

RICH, C., SIDNER, C., AND LESH, N. 2001. COLLAGEN: Applying collaborative discourse theory. *AI Magazine 22,* 4, 15–25.

RUDNICKY, A. I., BENNETT, C., BLACK, A., CHOTOMONGCOL, A., LENZO, K., OH, A., AND SINGH, R. 2000. Task and Domain Specific Modelling in the Carnegie Mellon Communicator System. In *Proc. ICSLP*. 130–134.

SENEFF, S. 2002. Response planning and generation in the MERCURY flight reservation system. *Computer Speech and Language 16,* 3–4, 283–312.

SMOLA, A. J. AND SCHÖLKOPF, B. 1998. A tutorial on support vector regression. NeuroCOLT2 Technical Report (NC2-TR-1998-030).

SPARKS, R., MEISKEY, L., AND BRUNNER, H. 1994. An object-oriented approach to dialogue management in spoken language systems. In *Proc. SIGCHI*. 211–217.

STURM, J., DEN OS, E., AND BOVES, L. 1999. Dialogue management in the Dutch ARISE train timetable information system. In *Proc. Eurospeech*. 1419–1422.

TAKANO, S., TANAKA, K., MIZUNO, H., ABE, M., AND NAKAJIMA, S. 2001. A Japanese TTS Syntem Based on Multi-form Units and a Speech Modification Algorithm with Harmonics Reconstruction. *IEEE Transactions on Speech and Audio Processing 9,* 1, 3–10.

VAPNIK, V. 1995. *The Nature of Statistical Learnig Theory*. Springer.

WALKER, M., KAMM, C., AND LITMAN, D. 2000. Towards developing general models of usability with PARADISE. *Natural Language Engineering: Special Issue on Best Practice in Spoken Dialogue Systems. 6*, 363–377.

WALKER, M. A., LITMAN, D. J., KAMM, C. A., AND ABELLA, A. 1997. PARADISE: A Framework for Evaluating Spoken Dialogue Agents. 271–280.

WANG, Y. AND WITTEN, I. H. 1997. Induction of model trees for predicting continuous classes. *Proc. European Conference on Machine Learning Poster Papers*, 128–137.

WITTEN, I. H. AND FRANK, E. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

ZUE, V. W. AND GLASS, J. R. 2000. Conversational interfaces: Advances and challenges. *Proc. IEEE 88,* 8, 1166–1180.