Crowdsourcing for Spoken Dialog Systems Evaluation

Zhaojun Yang, Gina-Anne Levow and Helen M. Meng

University of Southern California, University of Washington and The Chinese University of Hong Kong

1.1 Introduction

A spoken dialog system (SDS) is a computer system which supports human-computer conversations in specific knowledge domains. It integrates technologies including speech recognition, natural language understanding, dialog modeling, language generation, and text-to-speech synthesis. Advances in speech and language technologies have made SDS an important research area and have brought about systems in a wide variety of applications, such as flight information (Hirschman et al. (1993)), bus schedule inquiries (Raux et al. (2005)), stock market information delivery (Meng et al. (2004a)), tourist information (Wu et al. (2006)) and student tutoring (Litman et al. (2006)). To facilitate the development of SDS and compare the performance of different systems, it is necessary to conduct SDS evaluation with appropriate methodologies.

A typical SDS architecture is illustrated in Figure 1.1. This implements a dialog interaction between two interlocutors, which consists of a series of dialog turns. A spoken dialog turn is a process in which one participant A utters something to the other B, and B interprets A’s utterance and then responds accordingly.

As illustrated in Figure 1.1, the first step for the system is to recognize the user’s speech with automatic speech recognition and interpret the underlying meaning with natural language understanding technologies. This involves extracting the user’s communicative (and informational) goal and inferring the appropriate follow up actions and responses. Language understanding involves a variety of methods, such as the use of parsers and grammars (Seneff (1992); Ward and Issar (1994)), belief networks (Meng et al. (1999, 2004b)), etc. The dialog model is the principal component of a dialog system which maintains the history of the dialog, decides which action is appropriate based on language understanding, and controls the dialog flow. A dialog model typically incorporates dialog states, state transitions, and a dialog policy. A dialog state represent the results of performing system actions in previous states; state transitions allow dialogs to move forward; and the dialog policy determines how
Crowdsourcing for Spoken Dialog Systems Evaluation

Figure 1.1  A typical spoken dialog turn between the system and the user.

to map dialog states into system actions (McTear (1998)). After the dialog model decides upon the most appropriate response, the system needs to convey its response to the user in audio form. This is achieved with language generation (Baptist and Seneff (2000); Cole et al. (1997)) and text-to-speech (TTS) synthesis technologies (Taylor (2009)).

SDSs are becoming increasingly pervasive in supporting information access by the masses. This calls for sound strategies in evaluating, comparing, and predicting the design and performance of such systems. Therefore, developing principled ways of evaluating an SDS has become a hot research area. Generally, SDS evaluation can be categorized into component-based and holistic approaches.

The component-based approach covers the performance of individual components such as correctness of speech recognition, ability to understand natural language, appropriateness of response generation, as well as the naturalness of the synthetic speech in conveying the responses. A thorough evaluation of an SDS needs to consider all relevant evaluation metrics covering the functionalities for all the system components (Mölle (2005)). As a result, different kinds of evaluation metrics have been proposed in previous work, such as query density and concept efficiency for measuring the system’s understanding ability (Glass et al. (2000)). The metrics have also been classified according to their functionalities into five categories: dialog-related, meta-communication-related, cooperativity-related, task-related, and speech-input-related (Mölle (2005)). Component-based evaluation may become challenging as the complexities of systems increase and their components become integrated in more intricate ways. This is especially true when we need to compare two systems with different components, or when some metrics are in conflict with one another. For example, we may have two systems A and B, where A uses explicit confirmations of the user’s information, and has dialogs with a longer dialog length but a higher task completion rate. B uses implicit confirmations, leading to dialogs with a shorter dialog length but a lower task completion rate. In this situation, we may face a dilemma where A has better effectiveness but B has better efficiency.

In contrast, holistic evaluation, which assesses not only individual components but also the integrated performance of an SDS, may be more appropriate. It involves the perceived level of system usability, system intelligence, and error recovery capabilities by considering
the system in its entirety (McTear (2002)). It also needs to cover the wide variety of users’ impressions (user judgments) relating to all dimensions of the quality of an SDS (Bonneau-Maynard et al. (2000)). The ultimate objective of an SDS is to satisfy the demands of real users. Therefore, user satisfaction is considered the most important criterion for system performance among all user impressions (Moller and Skowronek (2003)).

Many evaluation methods have been developed in recent years. One popular method of measuring user judgments is to invite subjects to fill out a questionnaire after they interact with an SDS. The questionnaire often involves perceptions of many aspects of the system, such as task completion and user satisfaction. In spite of its popularity, this traditional approach has some disadvantages. First, it is a costly and time-consuming process. Moreover, due to limitations in resources, this approach is often constrained to a small number of evaluators whose feedback may not be statistically representative of the large user population that can access the SDS. Furthermore, in some situations where the system has already been deployed, it is often difficult to ask real users to patiently complete an evaluation survey.

Another popular evaluation method, the PARADISE framework, has been proposed for automatic inference of overall user satisfaction of unrated dialogs (Walker et al. (1997)). It assumes that the overall performance of an SDS can be described in terms of a linear regression model of a set of dialog metrics, to maximize this usability-related measure (Walker et al. (1998)). The trained model can explicitly demonstrate which factors have significant contributions to user satisfaction. Nevertheless, such frameworks still need evaluated dialogs to train the predictive model, whose performance may also be determined by the amount of graded data (Engelbrecht and Möller (2007)).

The challenges in SDS evaluation motivated us to explore the use of crowdsourcing as a potentially effective and efficient paradigm. We note that crowdsourcing technologies have been widely applied by many researchers to collect (See Chapters ?? and ??), transcribe (See Chapter ??), and annotate speech and language data in recent years (McGraw et al. (2010); Novotney and Callison-Burch (2010); Snow et al. (2008)). Crowdsourcing refers to outsourcing a task to a crowd of people. Unlike using the traditional method in which data is manually labeled by experts or trained people, tasks can be completed with crowdsourcing in a cost-effective, efficient, and flexible manner. We feel that user judgments for SDS evaluation can also be collected by using crowdsourcing instead of user experiments. In this chapter, we will describe our work in developing a crowdsourcing methodology for SDS evaluation through Amazon Mechanical Turk (MTurk) (http://www.mturk.com), a popular crowdsourcing marketplace that makes use of human intelligence online to perform tasks which cannot be completed entirely by computer programs.

This chapter is organized as follows: In section 1.2, we discuss prior work in crowdsourcing for dialog and speech assessment. In section 1.3, we provide an overview of related work on SDS evaluation. The remainder of the chapter details a specific approach to and analysis of crowdsourcing for SDS evaluation, to serve as an illustrative example. Section 1.4 presents information about the experimental corpus and how automatic dialog classification is done. Section 1.5 presents the methodology in the use of crowdsourcing for collecting user judgments on SDSs. We design two types of tasks – the first targets rapid rating of a large number of dialogs with regard to some dimensions of the SDS’s performance, and the second aims to assess the reliability of workers in crowdsourcing. To address the particular challenges of crowdsourcing, we structure these tasks to elicit judgments on a representative sample of dialogs, support semi-automatic task validation, and ensure task
clarity. To control the quality of ratings from crowdsourcing, we also develop and present a set of approval rules. Section 1.6 presents an analysis of collected results, which demonstrates that the crowdsourcing method for the collection of user judgments is efficient, flexible and cost-effective. The comparison of annotations between experts and workers on SDSs shows a high level of agreement between the two groups, which supports the reliability of crowdsourcing. Finally, this chapter concludes with Section 1.7.

1.2 Prior Work on Crowdsourcing: Dialog and Speech Assessment

1.2.1 Prior Work on Crowdsourcing for Dialog Systems

Some recent approaches have explored the use of crowdsourcing in the development of dialog systems. These approaches have aimed to exploit the crowd to create or enhance text-based dialog interactions. Bessho et al. (2012) investigated the automatic generation of replies to tweets on the Twitter microblog. Their approach employed a similarity based technique using a database of tweet-reply pairs. For a new input, the most similar tweet in the database is identified and its corresponding reply is generated as the response. If no sufficiently similar tweet is present, the tweet is sent to a ‘crowd’ in real time, and if a reply is provided within a given time window, that crowdsourced reply is used as the system response.

DePalma et al. (2011) provide an example of the use of crowdsourcing to develop models of Human-Robot dialog (HRI). They describe the use of crowdsourced interactions from a game called ‘Mars Escape’ in which players take the roles of an astronaut and a robot doing a collaborative task. Actions and communication via text-based chat are logged. They also planned a real-world variant with a robot in the Boston Museum of Science. These crowdsourced interactions provided a much richer set of dialog behaviors than are typically present in hand-coded HRI systems.

1.2.2 Prior Work on Crowdsourcing for Speech Assessment

Crowdsourcing has also been employed for the collection of judgments of speech quality, particularly in the field of speech synthesis. Ribeiro et al. (2011) describes a framework for collection of Mean Opinion Scores (MOS) through crowdsourcing, applying the basic approach to assessment of synthesized speech, as well as of image quality and region of interest determination. The tasks required workers to listen to speech samples and provide scores from 1 to 5. They streamlined the tasks through interface design, such as the use of radio buttons, and by dropping pre-qualification tasks. Instead they developed strategies for quickly rejecting poor or cheating workers, by rejecting incomplete tasks, tasks completed too quickly, and workers with low correlation coefficients. Known good and bad samples were also included to enable automatic validation. Lastly, they aimed to recruit and retain good workers, by providing bonuses for completing more tasks and maintaining high quality, as measured by correlation coefficient. This approach allowed high throughput at a relatively low cost, important since MOS testing relies on large numbers of subjects for statistical power.

Wolters et al. (2010) describe the crowdsourcing of speech synthesis assessment through an intelligibility task. Workers are asked to listen to utterances and transcribe them word-for-word. Since the utterance text itself is available, worker input can be evaluated with respect
to that gold standard. Workers with too many errors were classified as cheaters.

Lastly, Buchholz and Latorre (2011) present the crowdsourcing of speech synthesis assessment through a preference task. Workers are asked to listen to pairs of utterances and indicate which they prefer. To validate worker input, they employed several checks: transitivity of ranking as a pseudo-gold standard, timing of audio play relative to preference entry, and distribution of ranks. They also used CrowdFlower’s geolocation service as a nativeness filter. They compared crowdsourced assessments to those of locally recruited researchers. They found that despite the presence of some systematic cheaters, the two groups produced similar preference trends, though the preferences were stronger for the workers, possibly due to differences in nativeness or task wording. Additional details on crowdsourcing and speech synthesis can be found in ??.

1.3 Prior Work in SDS Evaluation

As introduced earlier, evaluation plays a critically important role in the design and development of SDS. The performance of an SDS can be measured in a multitude of ways, such as task success, the number of dialog turns, speech recognition accuracy, system response delay, naturalness of the output speech, consistency with the user’s expectations, and cooperativeness of the system (Möller (2005)). These evaluation metrics are usually categorized into subjective and objective metrics. Subjective metrics, which reflect the users’ perceptions of the quality of an SDS, are often obtained from real or test users. Objective metrics, which quantify system behavior during the interaction and the performance of the components of an SDS, can be extracted automatically or labeled manually from the user-system interactions by expert evaluators. The objective metrics are also called interaction metrics in Möller (2005).

1.3.1 Subjective User Judgments

Since subjective metrics mostly rely on user judgments of system quality, distributing questionnaires to users before or after interaction with an SDS is an effective way to collect quantifiable user judgments. Developing a reliable and valid questionnaire for subjective judgment collection involves many design considerations. The SASSI questionnaire (Subjective Assessment of Speech System Interfaces) is designed for subjective assessment of speech-based systems (Hone and Graham (2001)). SASSI consists of 50 items (or statements), and each is rated by users on a 7-point scale: strongly agree, agree, slightly agree, neutral, slightly disagree, disagree and strongly disagree. A factor analysis of the collected data from 226 completed questionnaires suggests that six main factors contribute to a user’s subjective perceptions of speech-based systems, i.e., perceived system response accuracy, likeability, cognitive demand, annoyance, habitability, and speed.

The International Telecommunication Union (ITU) recommendation proposed another list of questions for the evaluation of SDS in telephone services (Rec (2003)). Three types of questionnaires are distinguished in the recommendation. Type 1 questionnaires are intended to collect the user’s background information and are distributed at the beginning of an evaluation experiment. Type 2 questionnaires are related to the user-system interactions. Type 3 questionnaires are about the users’ overall impression of system quality. A list of topics are proposed for each type of questionnaire, and the statements are rated on a 5-point scale.
1.3.2 Interaction Metrics

In contrast to subjective judgments of system performance, interaction metrics can easily quantify the ability of a system or its components to perform the designed functions. Such information is obtained from the log files which record the interactions between the system and its users. Certain metrics that provide an overview of user-system interactions can be automatically extracted from log files, such as dialog duration, recognition confidence scores, etc. Other metrics are related to the content of the interactions and are usually manually labeled by experts or trained annotators, such as the accuracy of understanding, task success, etc.

In recent decades, many metrics have been identified to measure the functionalities of a system and its components. Early metrics are for individual components, such as the speech recognizer and language understanding component. Commonly used metrics are Word Accuracy (WA), Sentence Accuracy (SA), Concept Accuracy (CA), Query Density (QD), Concept Efficiency (CE), etc. Later, metrics for whole systems have been developed, including Task Success (TS) to measure the extent to which the system achieves the task, number of dialog turns for measuring the dialog cost, or Contextual Appropriateness (CA) for measuring the degree to which the system provides an appropriate response (Glass et al. (2000); McTear (1998)).

Based on the literature of interaction metrics, Möller et al. summarize a set of metrics for SDSs evaluation and classify them into five categories:

- **Dialog- and communication-related category**: Metrics about the overall dialog, such as overall dialog duration, dialog turns, or average number of words per system turn during the dialog, etc.
- **Meta-communication-related category**: Metrics describing the recognition and understanding capabilities, such as number of help requests, number of barge-in attempts from the user, etc.
- **Cooperativity-related category**: Metrics about the cooperativity of system actions (responses). The contextual appropriateness of system responses directly measures cooperativity, which is often judged by experts based on Grice’s maxims.
- **Task-related category**: Successful task completion is a key requirement for task-oriented systems. Möller defined seven levels of task success, i.e., success by providing a completely right answer; success with constraints from user, or from system, or from both user and system; success by determining that no solutions exist; failure resulting from user’s non-cooperative behavior or system’s inappropriate response.
- **Speech-input-related category**: Metrics about the capability of systems to recognize the input speech and to understand the meaning of inputs. Commonly used metrics are WA, SA, or CA as introduced above.

This categorization and the metrics in each category have been incorporated in the ITU recommendation (series Rec (2005)).

1.3.3 The PARADISE Framework

PARADISE (PARAdigm for DIalogue System Evaluation) is a general framework for evaluating and comparing the performance of spoken dialog systems. It quantifies the
Figure 1.2 The PARADISE structure of objectives for spoken dialog performance (Walker et al. (1997)).

contributions of different system properties to system usability and supports the development of predictive models of system performance (Walker et al. (1997)). The PARADISE framework uses decision-theoretic methods to relate a collection of dialog metrics to the system’s overall performance and determine the significant contributors. The PARADISE performance model is shown in Figure 1.2. The overall performance is correlated with user satisfaction, and the primary objective of a system is to maximize user satisfaction. This objective can be further decoupled into two sub-objectives: maximizing task success and minimizing dialog cost, based on the assumption that task success and dialog cost are two main types of contributors to user satisfaction. In the original PARADISE framework, task success is measured with the use of the Kappa coefficient and attribute-value matrix (AVM). Dialog costs can be categorized into two types: dialog efficiency and quality. Dialog efficiency is represented by the number of dialog turns or the dialog duration, while dialog quality is measured in terms of the appropriateness of system response, system repair ratio, etc.

The PARADISE framework posits that the objective structure in Figure 1.2 can be realized by building a performance model through multivariate linear regression with user satisfaction as the target and the dialog metrics of task success, dialog efficiency and quality as predictors. Building the performance model requires a dialog corpus be collected through controlled user experiments during which users subjectively rate their satisfaction. Moreover, the predictors of the model, i.e., the dialog metrics, can be either automatically extracted from dialog log
files or manually labeled by experts. Based on these illustrations, the performance model of an SDS is defined below:

\[ S_u = (\alpha \ast N(\kappa)) - \sum_{i=1}^{n} w_i \ast N(c_i), \tag{1.1} \]

where \( S_u \) is system performance correlated with user satisfaction, \( \kappa \) is a measure for task success, \( c_i \) is a measure for dialog cost, \( \alpha \) is a weight on \( \kappa \), \( w_i \) is a weight on \( c_i \), and \( N(\cdot) \) is a z-score normalization function (Cohen (1995)). Both \( \kappa \) and \( c_i \) can be represented as dialog measures \( m \), and Equation 1.1 is transformed into a simpler one:

\[ S_u = \sum w_i \ast N(m_i). \tag{1.2} \]

Since the dialog measures have been normalized into the same scale by \( N(\cdot) \), the weight \( w_i \) reflects the relative contribution of the corresponding measure \( m_i \) to user satisfaction.

By applying the performance model, values of user satisfaction for SDSs are directly predicted from a suite of dialog metrics which are simply extracted from dialogs, without the need to conduct user experiments to assess user satisfaction. In addition, system developers can directly figure out which system components have a greater impact on user satisfaction by observing the coefficients of dialog metrics in the performance model, so that they can focus on improving the performance of those ‘important’ components. In this way, an efficient system design process is ensured, i.e., focusing on high service quality for the end user.

PARADISE has been widely applied in evaluating many SDSs, such as the ITSPOKE tutoring system (Forbes-Riley and Litman (2006)) and DARPA Communicator (Walker et al. (2001)). It is applied to test different dialog models with two systems ELVIS and TOOT in (Walker et al. (1998)). The ELVIS experiments tested different initiative dialog models, while the TOOT experiments tested models with different information presentation policies. Regression functions are derived for each model to analyze their performance. Kamm et al. discusses the generalization ability of PARADISE across three different systems (Kamm et al. (1999)). Walker et al. study its generalization across different user populations and find that it does not generalize well from novice users to expert users (Walker et al. (2000)). Additionally, researchers have extended PARADISE to evaluate multimodal systems (Beringer et al. (2002); Dybkjaer et al. (2004)).

1.3.4 Alternative Approach to Crowdsourcing for SDS Evaluation

Aside from the approach presented later in this chapter, Jurcicek et al. (2011) present the only other approach to date for evaluating spoken dialog systems themselves through crowdsourcing. In their case, the task requires workers to interact with an SDS to perform a prespecified task and complete a brief questionnaire focused on task completion, asking if they obtained the information sought. Due to technical difficulties with enabling the SDS interaction through the MTurk interface itself, workers interacted with the SDS over the telephone and were given a code to enter in the crowdsourcing interface as part of task verification. Task performance was validated by comparison with the goals specified in the assigned task and those inferred from the interaction logs. The crowdsourced results were compared with those from locally recruited users across different dialog system models and found to be largely consistent.
Table 1.1 An example of the Let’s Go! dialogs.

| S: Welcome to the CMU Let’s Go bus information system |
| S: What can I do for you? |
| U: When is the next 64A from Forbes and Murray to East Liberty? |
| S: The 64A. Did I get that right? |
| U: Yes. |
| S: Leaving from Forbes and Murray. Correct? |
| U: Yes. |
| S: Going to East Liberty. Am I right? |
| U: Yes |
| S: I think you want the next bus. |
| U: Yes |
| S: Hold on. I’ll look that up. |
| S: The next 64A leaves Murray Avenue at Forbes at 10:32p.m. and arrives at Highland Avenue at Penn Avenue at 10:55p.m. |

1.4 Experimental Corpus and Automatic Dialog Classification

The corpus used in this case study contains about 50,000 dialogs drawn from the log files collected through the Let’s Go! dialog system developed by Carnegie Mellon University (Raux et al. (2005)). This system provides bus schedule information for Pittsburgh residents when the customer service hotline is not staffed. Input utterances from users are transcribed automatically by automatic speech recognition (ASR). Table 1.1 shows an example of the Let’s Go! dialogs.

We develop a simple characterization of the dialogs in the corpus. We survey the dialog logs and identify several cue phrases that enable us to develop a set of heuristics to automatically classify the dialog logs into five categories in terms of task success. The dialog logs are tracked from the end towards the beginning with a set of ordered rules:

1. Dialogs with no more than 3 turns are classified as 'TooShort.'
2. Dialogs with the cue phrases, 'Start a new query', 'New query', etc., in the user turns are classified as 'MultiTask.'
3. Dialogs with the cue phrase '...result' in the system turns are classified as 'TaskComplete.'
4. Dialogs with the cue phrase 'I’m sorry....' in the system turns are classified as 'OutofScope.'
5. If none of the above rules apply, the dialog is classified as 'TaskIncomplete.'

Example dialogs in each of these five categories are shown in Figure 1.2. In TooShort dialogs, the user usually hung up after only a single utterance, indicating that the user didn’t really want to use the system. Therefore, such dialogs are separated from the TaskIncomplete

---

1This cue phrase is observed in the grammar tags of the Let’s Go! System’s dialog logs.
Table 1.2  Samples of the Let’s Go! dialogs in the different categories of automatic classification.

Case 1: Task Complete
S: Welcome to the CMU Let’s Go bus information
S: What can I do for you?
U: When is the next 64A from Forbes and Murray to East Liberty?
S: The 64A. Did I get that right?
U: Yes.
S: Leaving from Forbes and Murray. Correct?
U: Yes.
S: Going to East Liberty. Am I right?
U: Yes.
S: I think you want the next bus.
U: Yes.
S: Hold on. I’ll look that up.
S: The next 64A leaves Murray Avenue at Forbes at 10:32pm and arrives at Highland Avenue at Penn Avenue at 10:55pm.

Case 2: TooShort
S: Welcome to the CMU Let’s Go bus information
S: What can I do for you?
U: OAKLAND .ANY. .FIND. .SOUTH SIDE. WANT.

Case 3: MultiTask
...
S: There is a 28X leaving THE AIRPORT...
U: NEW QUERY
S: Okay, let’s start from the beginning.
...

Case 4: OutofScope
...
S: I’m sorry, I don’t know any routes...

Case 5: TaskIncomplete
...
S: Leave at 8 a.m. Is this correct?
U: YES
S: Hold on, I’ll look that up.
ones. The distribution of the classification results is shown in Figure 1.3. About 35% of the dialogs are in the TaskComplete category, while the remaining categories share comparable percentages. As will be described later, this rough classification is used to determine the proportion of the different types of dialogs that are uploaded to MTurk for crowdsourcing. The rough classification also provides a means by which we can verify whether the human raters on the crowdsourcing platform have put effort into rating our dialogs, as opposed to providing random (and unreliable) inputs. We will elaborate on these points in the following sections.

![Distribution of five dialog categories based on automatic, heuristics-based classification.](image)

**Figure 1.3** Distribution of the five dialog categories based on the automatic, heuristics-based classification.

### 1.5 Collecting User Judgments on Spoken Dialogs with Crowdsourcing

As introduced in Section 1.3, the subjective user judgments on SDSs are considered the most important criterion for SDS performance. However, user judgments are usually measured by asking subjects to fill out a questionnaire after they interact with an SDS, which is a costly, tedious and time-consuming process. This section presents an approach to the use of crowdsourcing for collection of user judgments on SDSs (Yang et al. (2010)). The objective of collecting user judgments with crowdsourcing is to have a large number of dialogs evaluated by a large (hence more statistically representative) group of people in an efficient and cost-effective manner, which is difficult to achieve using traditional methods. In the experiments detailed below, the crowdsourcing approach is implemented through Amazon Mechanical Turk (MTurk). MTurk was chosen based on its well-established status at the time of the experiments; however, the basic approach could be deployed on alternative platforms (See Chapter ?? for other candidates). The MTurk platform organizes work in the form of Human Intelligence Tasks, which we will refer to here as ‘tasks’ consistent with the terminology in Chapter ??.

A task is designed by the ‘Requester’ (i.e., the research team) and is completed by many 'Workers' (i.e., anyone who is interested in the task) over the Internet.
We describe a design methodology for two types of tasks – the first targets rapid rating of a large number of dialogs on several dimensions of SDS performance, and the second aims to assess the reliability of workers on the crowdsourcing platform through an assessment of interannotator agreement in ratings among different workers and between workers and experts. We structure the tasks to elicit ratings of a representative set of dialogs, to enable semi-automatic validation of worker submissions, and to achieve clear tasks. In addition, we develop and apply a set of approval rules, required to exclude submissions with nonsensical ratings, which would affect the overall quality of the ratings obtained. We describe the tasks and the approval process below.

1.5.1 Tasks for Dialog Evaluation

This type of task is designed to outsource the assessment of the SDS to workers. The assessment focuses on selected dimensions of performance regarding the SDS, based on a large number of dialogs selected from the logs. To achieve this goal, we have authored a set of questions that constitute the task in Table 1.3.

In Table 1.3, we include the explanation of the aim for each question, but this is not shown to the workers. These questions cover the user's confidence, perceived task completion, expected behavior, overall performance and categorization of task success. We chose these aspects for evaluation because they represent a range of important factors in system quality and because our data included only the textual transcription of dialogs, which made evaluation of other aspects of the systems, such as speech output quality, impractical. This final questionnaire was developed in the course of pilot experiments, in which questions which caused confusion among workers were revised and clarified. For example, the definition of task completion was particularly problematic and led us to the options in Q5 based on the ITU Recommendation (series Rec (2005)). We have purposely designed the questions in such a way that they can cross-validate each other (Q2 and Q5 both aim to assess task completion), which will be used for approval of ratings from crowdsourcing later.

Each task contains the text transcription of one dialog and the questionnaire in Table 1.3 for assessment by the workers, who are paid USD $0.05 for each task completed. We have uploaded 11,000 dialogs in total, including samples from the three major dialog categories and in proportions that follow the percentages obtained from the automatic classification, i.e., TaskComplete (55%), TaskIncomplete (27%), OutofScope (18%). TooShort and MultiTask dialogs are excluded from the task. The former is easily detectable as unsuccessful. The latter can be easily segmented into mono-task dialogs, which can then follow the three-way categorization (TaskComplete / TaskIncomplete / OutofScope) directly.

1.5.2 Tasks for Interannotator Agreement

This type of task is an extension of those in Section 1.5.1 and is designed to assess the reliability of workers in crowdsourcing through interannotator agreement. Each task includes the text transcriptions of 30 selected Let’s Go! dialogs drawn from log files (10 dialogs from the categories of TaskComplete, TaskIncomplete and OutofScope respectively). Each dialog is associated with the questionnaire in Table 1.3. Workers are paid USD $1.5 for each task completed. Altogether, we have 3 groups of workers (each with 16 individuals) rating two sets.
**Table 1.3** Questions constituting the task on Dialog Evaluation (Q: Question, Opt: Options). The questionnaire covers the user’s confidence, the perceived task completion, the expected behavior, the overall performance and the categorization of task success.

<table>
<thead>
<tr>
<th>Q1</th>
<th>Do you think you understand from the dialog what the user wanted?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opt</td>
<td>1) No clue  2) A little bit  3) Somewhat  4) Mostly  5) Entirely</td>
</tr>
<tr>
<td>Aim</td>
<td>elicit the worker’s confidence in his/her ratings.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q2</th>
<th>Do you think the system is successful in providing the information that the user wanted?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opt</td>
<td>1) Entirely unsuccessful  2) Mostly unsuccessful  3) Half successful/unsuccesful  4) Mostly successful  5) Entirely successful</td>
</tr>
<tr>
<td>Aim</td>
<td>elicit the worker’s perception of whether the dialog has fulfilled the informational goal of the user.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q3</th>
<th>Does the system work the way you expect it?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opt</td>
<td>1) Not at all  2) Barely  3) Somewhat  4) Almost  5) Completely</td>
</tr>
<tr>
<td>Aim</td>
<td>elicit the worker’s impression of whether the dialog flow suits general expectations.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q4</th>
<th>Overall, do you think that this is a good system?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opt</td>
<td>1) Very poor  2) Poor  3) Fair  4) Good  5) Very good</td>
</tr>
<tr>
<td>Aim</td>
<td>elicit the worker’s overall impression of the SDS.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q5</th>
<th>What category do you think the dialog belongs to?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opt</td>
<td>1) TS:Fu – Failed because of the user behavior, due to non-cooperative user behavior  2) TS:Fs – Failed because of the system behavior, due to system inadequacies  3) TS:SN – Succeeded in spotting that no solution exists  4) TS:CsCu – Succeeded with constraint relaxation both from the system and from the user  5) TS:Cu – Succeeded with constraint relaxation by the user  6) TS:Cs – Succeeded with constraint relaxation by system  7) TS:S – Succeeded</td>
</tr>
<tr>
<td>Aim</td>
<td>elicit the worker’s impression of whether the dialog reflects task completion.</td>
</tr>
</tbody>
</table>
of dialogs (each with 30 dialogs). Groups 1 and 2 evaluate the first set of dialogs, while Group 3 evaluate the second set. In this way, we can assess whether the interannotator agreement varies across different raters and different dialogs.

1.5.3 Approval of Ratings

It is important to verify the quality of inputs from workers. Since the quality of the ratings directly impacts the credibility of the SDS evaluation, some basic rules have to be set to ensure the workers are devoting effort to their tasks and to guarantee the reliability of ratings, in addition to the qualification requirement preset for the workers. We have developed an approval mechanism, as follows:

R1. We reject tasks for which the working time is less than 15 seconds, since we feel that careful (and thus high quality) ratings cannot be completed within such a short period.

R2. If a worker completes a large number of tasks (e.g., over 20) but provides identical answers for all of them, his/her work will be rejected.

R3. Approval requires consistency between the answers to related questions (Q2 and Q5). Consistency is based on four main heuristics:

- Answers to Q2 being "Entirely successful" or "Mostly successful" can go with answers to Q5 being TS:S, TS:CS, TS:Cu, or TS:CsCu.
- Answers to Q2 being "Entirely unsuccessful" or "Mostly unsuccessful" can go with answers to Q5 being TS:Fs or TS:Fu.
- The answer to Q2 being "Half unsuccessful / successful" can go with any answer in Question 5.
- The answer to Q5 being TS:SN can go with any answer to Q2.

R4. Approval requires consistency between the answers to Q5 and the automatic classification of the dialogs (see Section 1.4). In particular, the heuristics are:

- TaskComplete can match with TS:S, TS:Cs, TS:Cu and TS:CsCu.
- TaskIncomplete can match with TS:Fs and TS:Fu.
- OutofScope can match with TS:SN.

R5. If these above heuristics are not satisfied, the dialog will be checked carefully. Random (incorrect) ratings are rejected. However, we have approved some ambiguous cases, as they will be explained in Section 1.6.2.

1.6 Collected Data and Analysis

This section describes the collected data, as well as presents the approval rates and the feedback from the workers. We also present an analysis of the collected data in terms of consistency among ratings and interannotator agreement.
11,000 dialogs were rated by around 700 online workers in 45 days. Three persons in our team completed the verification of the rated tasks and approved 8,394 of them. The total expenditure paid to the workers is USD $350. Approval rates for each dialog category, i.e., TaskComplete, TaskIncomplete and OutofScope, are shown in Figure 1.4 respectively. OutofScope is the highest because some workers consider a task to be successful if they think that the absence of the information is due to the database but not the ability of the system. Others consider such cases to be failures since the system does not provide the requested information for the users. We approve either decision from the workers.

Rejected dialogs led to some controversies. Some apologized for their errors and others complained about the rejections. We received feedback from the workers concerned, many of which are useful to help enhance our understanding of SDS evaluations. Here we list some typical comments associated with their implications as follows.

- The system does not provide the exact information that the user wanted although it provides some related results. *(Retrieval result from database is a vital aspect of SDS performance.)*

- The system’s understanding ability is very important, so good understanding may lead workers to choose task success even if the system does not provide any information to the user. *(Good language understanding ability plays an important role in improving user satisfaction.)*

- The system succeeds in providing a result based on the user’s initial input but fails to follow up with the user’s updated information. *(Timely updates of the dialog state impacts users’ perception of SDS performance positively and greatly.)*
1.6.2 Consistency between Automatic Dialog Classification and Manual Ratings

To assess the quality of ratings from crowdsourcing, we investigate the consistency between automatic dialog classification (see Section 1.4) and the manual ratings from workers (with respect to Questions 2 and 5) based on the approved tasks of about 8,000 dialogs.

According to the approval rules, the tasks whose ratings of Q5 are not consistent with the automatic classification will be rejected (see R4 in Section 1.5.3). However, as mentioned in Section 1.5.3, we still approve some ambiguous dialogs for which it is difficult to determine task success. The manual ratings of Q5 for the ambiguous dialogs do not agree with automatic classification. Table 1.4 shows an example, where the dialog is terminated midway. Some workers regard the dialog as successful (in Question 5), because they feel that the system would have completed the task had the user not hung up. Nevertheless, since the dialog does not have the necessary cue phrase (i.e., '..result..') in the log, it is automatically classified as TaskIncomplete. Percentages of ambiguous dialogs over the total approved ones in each category of TaskComplete, TaskIncomplete and OutOfScope are shown in Figure 1.5 respectively. Note that a higher percentage of ambiguous dialogs are approved for the OutOfScope category, mainly due to workers’ diverse understandings of such dialogs, as we have discussed in Section 1.6.1.

Moreover, Figure 1.6 plots the mean answer scores of the two questions for approved dialogs in each of the three categories, where a higher score maps to a higher level of success, namely:

- Answer scores to Q2 range from 0 for 'Entirely unsuccessful' to 1 for 'Entirely successful'.
Table 1.4 An ambiguous dialog from TaskIncomplete category.

<table>
<thead>
<tr>
<th>System</th>
<th>User</th>
<th>System</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome to the CMU Let’s Go bus information system.</td>
<td>TWENTY. EIGHT. AT. IN. DOWNTOWN. FIFTY. AIRPORT.</td>
<td>Leave from AIRPORT. Is this correct?</td>
<td>FORM DOWNTOW</td>
</tr>
<tr>
<td>To get help at any time, just say Help or press zero.</td>
<td></td>
<td>Leave from DOWNTOWN. Is this correct?</td>
<td>YES</td>
</tr>
<tr>
<td>What can I do for you?</td>
<td>Where are you going to?</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>System: Leave from AIRPORT. Is this correct?</td>
<td>System: You want to go to AIRPORT. Is this correct?</td>
<td>System: What time are you leaving?</td>
<td>System: leave at 8 a.m. Is this correct?</td>
</tr>
<tr>
<td>System: Where are you going to?</td>
<td>User: TO AIRPORT</td>
<td>User: EIGHT A M</td>
<td>User: YES</td>
</tr>
</tbody>
</table>

- Answer scores to Q5 range from 0 for 'TS:Fu' to 1 for 'TS:S'.

The scores to Q2 and Q5 have been normalized to the same range from 0 to 1.

Generally, although some ambiguous dialogs are approved, we still observe reasonable agreement, i.e., the dialogs automatically classified as TaskComplete receive high scores from the Workers, those automatically classified as Task-Incomplete receive low scores, and those in the OutofScope category receive middling scores. Such consistency verifies the reliability of the approved ratings from crowdsourcing to some extent.

1.6.3 Interannotator Agreement Among Workers

As mentioned earlier, the second type of task (see Section 1.5.2) is designed to assess the level of interannotator agreement (ITA) among the workers. We adopt Cohen’s weighted kappa measure which is often applied to ordinal categories,

$$W_{weighted} = \frac{\sum_{i=1}^{c} \sum_{j=1}^{c} w_{ij}(n_{ij}/N - n_i n_j/N^2)}{1 - \sum_{i=1}^{c} \sum_{j=1}^{c} w_{ij} n_i n_j/N^2},$$  \hspace{1cm} (1.3)

where $c$ is the number of categories (i.e., answer options for each question here, $c = 5$ for Q1-Q4 and $c = 7$ for Q5), $w_{ij} = 1 - \frac{(i-j)^2}{c^2}$, $n_{ij}$ is the element in the observed matrix, $n_i = \sum_j n_{ij}$ and $n_j = \sum_i n_{ij}$. Details can be found in (Shoukri (2004)). A higher weighted kappa value indicates a higher interannotator agreement.
Recall that we have three groups of workers rating two sets of dialogs. These ratings are accepted directly and do not undergo the approval process. For any pair of workers in each group, we compute the weighted kappa value for each question. We then compute the mean weighted kappa value for each question over the entire group. Results are shown in Figure 1.7. Despite the fact that groups 1 and 2 evaluated the same dialog set, while group 3 evaluated a different dialog set, the three plots remain close, which illustrates that the interannotator agreement for each question remains stable across different raters and different dialogs. In particular, Q5 (categorization of task success) achieves mean weighted kappa values above 0.6 and Q2 (perceived task completion) achieves reasonable values above 0.4, which is indicative of a moderate level of agreement (Landis and Koch (1977)). Q2 and Q5 related to task success and thus elicit relatively objective ratings from reliable raters, so the moderate and stable agreement partially shows the reliability of workers and provides support for the utilization of crowdsourcing as a judgment collection mechanism. On the other hand, Q3 (expected behavior) and Q4 (overall impression on system performance) have low values below 0.3, which is indicative of a lack of agreement. This suggests that evaluation based on overall user satisfaction may be quite subjective and that it may be necessary to further refine questionnaires to better elicit such assessments. The low agreement in user satisfaction may lead to the low prediction accuracy for the evaluation model, which has been analyzed in (Engelbrech et al. (2009)).

Figure 1.7 also shows an interesting observation that for Q2 and Q5, the relative value gap between group 1 and 2 is larger than that between group 2 and 3. Intuitively, the values of interannotator agreement in group 1 and 2 should be closer, since workers in the two groups evaluate the same set of dialogs. We investigate the annotations of workers in each group. Results show that three workers in group 1 are obviously unreliable. For example, they rate the TaskComplete dialog in Table 1.5 as TaskIncomplete which contradicts with

Figure 1.6  The normalized mean scores of Q2 and Q5 for approved ratings in each category. A higher score maps to a higher level of task success.
the rates of most group members, while there are no such inconsistent inputs in the other two groups. These inconsistent annotations from the few unreliable workers lead to the lower interannotator agreement in group 1. Despite this difference, the level of interannotator agreement over the two groups is still the same. If we exclude the ratings from the three unreliable workers, the interannotator agreement of Q2 will increase from 0.444 to 0.542, and that of Q5 will increase from 0.621 to 0.698.

![Mean values of weighted Kappa of five questions for 3 groups. The weighted Kappa values are stable across different raters and different dialogs. Q2 and Q5 which can gain 'official' ratings from reliable raters have high levels of interannotator agreement.](image)

**Figure 1.7** Mean values of weighted Kappa of five questions for 3 groups. The weighted Kappa values are stable across different raters and different dialogs. Q2 and Q5 which can gain 'official' ratings from reliable raters have high levels of interannotator agreement.

We conceive of several factors that cause low interannotator agreement in overall user satisfaction. First, different users may emphasize different aspects of system performance, ranging from the system’s intelligence, task completion, dialog efficiency, etc. Second, raters with different levels of domain knowledge may have different expectations of the system. It may not be meaningful to compute an overall average score of user satisfaction across a diversity of users. Instead, we may consider the following possibilities:

- Evaluate SDSs along different dimensions individually.
- Evaluate SDSs in terms of different types of user queries, targeting different system functionalities.
- Evaluate SDSs based on different user groups with different levels of domain knowledge.
Table 1.5  A TaskComplete dialog from the dialog set in the task for interannotator agreement.

S: Welcome to the CMU Let’s Go bus information system.
S: To get help at any time, just say Help or press zero.
S: What can I do for you?
U: I’D LIKE TO GO FROM FORBES AND BIGELOW TO SQUIRREL HILL
S: Leaving from FORBES AND BIGELOW. Did I get that right?
U: YES
S: Right. Going to SQUIRREL HILL. Did I get that right?
U: YES
S: Okay. When do you want to travel?
U: NEXT AVAILABLE
S: When would you like to travel?
U: RIGHT NOW
S: You want the next bus. Is this correct?
U: YES
S: Just a minute. I’ll look that up.
S: The next 61A leaves FORBES AVENUE AT BIGELOW SCHENLEY DRIVE at 10:40 p.m. and arrives at FORBES AVENUE AT MURRAY at 10:49 p.m.

1.6.4 Interannotator Agreement on the Let’s Go! System

Section 1.6 has assessed the reliability of the SDS evaluation from workers using crowdsourcing by comparing the crowdsourced annotations to automatic classification and analyzing the interannotator agreement among workers. Nonetheless, there remains a key issue not resolved: whether the SDS evaluation by workers is as reliable as that by experts, in this case, researchers in the area of spoken dialog systems. In this section, we will investigate the level of agreement between the non-expert and expert annotations through two case studies.

The second type of task in Section 1.5.2 contains a set of 30 dialogs from the Let’s Go! dialog corpus, of which each 10 dialogs are from the categories of TaskComplete, TaskIncomplete and OutofScope. We have 4 experts evaluate the set of dialogs and compare their results with those of one group of 16 workers who evaluate the same set of dialogs in Section 1.5.2. The weighted kappa in Equation 1.3 is again employed to measure the interannotator agreement (ITA) (Shoukri (2004)).

For any pair of experts, we first compute the ITA value for each question. We then calculate the mean expert-expert (E-E) ITA for each question over all the pairs. Given an expert E, the ITA value of any pair of E and workers is calculated, and then the E-Workers (E-W) ITA is obtained by averaging the ITA values over all the E-Worker pairs. Intuitively, reliable workers are expected to have a high level of agreement with experts.

Table 1.6 shows E-W and E-E ITA values for each question. There’s a high level of agreement between experts and workers for question 2 and 5 (the weighted kappa values
are around 0.7) which are about classification of task completion and represent relatively objective ratings (Landis and Koch (1977)), and the $E$-$W$ ITA values approach those of $E$-$E$ ITA. This agreement indicates that workers perform well on these measures and the crowdsourced data could be considered reliable. On the other hand, similar to the results of ITA among workers in Section 1.6.3, Q3 (expected behavior) and Q4 (overall impression on system performance) have lower ITA values for both $E$-$W$ and $E$-$E$, which is indicative of lower agreement. Although agreement is slightly higher among the experts, it remains lower overall and suggests that evaluation based on overall user satisfaction may be quite subjective.

<table>
<thead>
<tr>
<th>Q1</th>
<th>$E_1$-$W$</th>
<th>$E_2$-$W$</th>
<th>$E_3$-$W$</th>
<th>$E_4$-$W$</th>
<th>$E$-$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5625</td>
<td>0.6107</td>
<td>0.5508</td>
<td>0.5525</td>
<td>0.6937</td>
</tr>
<tr>
<td>Q2</td>
<td>0.6744</td>
<td>0.7037</td>
<td>0.7160</td>
<td>0.7434</td>
<td>0.8709</td>
</tr>
<tr>
<td>Q3</td>
<td>0.2838</td>
<td>0.4368</td>
<td>0.2441</td>
<td>0.4214</td>
<td>0.5288</td>
</tr>
<tr>
<td>Q4</td>
<td>0.3686</td>
<td>0.4295</td>
<td>0.3188</td>
<td>0.4416</td>
<td>0.6240</td>
</tr>
<tr>
<td>Q5</td>
<td>0.7984</td>
<td>0.8242</td>
<td>0.8209</td>
<td>0.81</td>
<td>0.9435</td>
</tr>
</tbody>
</table>

### 1.6.5 Consistency Between Expert and Non-expert Annotations

The 2010 Spoken Dialog Challenge (SDC) was organized by the Dialog Research Center at CMU. The aim of this challenge was to realistically compare different spoken dialog systems on the same dialog task and make the collected dialogs from the systems available for the development of evaluation techniques.

In the 2010 SDC, there are four participating systems (the three competitors and the CMU reference system). All these systems provide Pittsburgh bus schedule information. Four dialog corpora was collected through the SDC controlled test which was conducted by asking subjects to call all the four systems. The organizers manually transcribed all the dialogues for each system and manually labeled task success for each dialog. A dialog is labeled as successful, if one piece of acceptable information is provided (Black et al. (2010)). We submit the text transcriptions of all the dialogues on MTurk using the task template in Section 1.5.1. Each dialog is evaluated by three workers. Task completion of a dialog is determined by the answers to Q5 with the use of majority vote. For example, if two or more workers choose one of the options of TS:S, TS:SCu, TS:SCu and TS:SCsCu, the dialog will be tagged as success.

We compare the task completion labels from workers with those from CMU experts. Several measures are defined for comparison. Given a dialog corpus $D = \{d_1, d_2, \cdots, d_N\}$, the crowdsourced labels of the corpus from workers are denoted as $L^W =$
\{l_1^W, l_2^W, \ldots, l_N^W\}, and labels from CMU experts are denoted as \(L^E = \{l_1^E, l_2^E, \ldots, l_N^E\}\), where \(l_i^W, l_i^E \in \{\text{Success}, \text{Failure}\}\). The first measure called success rate (SR) is defined in the following way,

\[
SR = \frac{\sum_{i=1}^{N} I(l_i = \text{Success})}{N},
\]

(1.4)

where \(I(\cdot)\) is:

\[
I(\omega) = \begin{cases} 
1 & \text{if } \omega = \text{true} \\
0 & \text{otherwise}.
\end{cases}
\]

(1.5)

The second measure of consistency rate (CR) is defined as below,

\[
CR = \frac{\sum_{i=1}^{N} I(l_i^W = l_i^E)}{N}.
\]

(1.6)

The third measure is consistent success rate (CSR):

\[
CSR = \frac{\sum_{i=1}^{N} I((l_i^W = \text{Success}) \land (l_i^E = \text{Success}))}{\sum_{i=1}^{N} I((l_i^W = \text{Success}) \lor (l_i^E = \text{Success}))}.
\]

(1.7)

Comparison results are shown in Table 1.7. The SR per system by the workers is quite close to that by the CMU experts. The CRs are all around 70%, even above 80% for system 3 and 4. The CSRs (around 80%) indicate that the ‘Success’ annotations from experts and non-experts have a large overlap. Table 1.7 demonstrates that worker judgments are generally consistent with expert judgments. This consistency also provides support for the utilization of crowdsourcing for collecting user judgments on dialogs.

| Table 1.7 Comparison of labels from CMU experts and workers. |
|-----------------|------|------|------|------|
|                 | Sys1 | Sys2 | Sys3 | Sys4 |
| Total dialogs   | 91   | 61   | 75   | 83   |
| SR (Workers)    | 56.1%| 36.1%| 86.7%| 78.3%|
| SR (CMU)       | 64.8%| 37.3%| 89.3%| 74.7%|
| CR             | 70.3%| 73.7%| 85.3%| 86.7%|
| CSR            | 88.1%| 80%  | 85.9%| 88.5%|

Next, we investigate the cases where the workers differ from the experts. We find that experts care more about the accuracy of the bus information that the system provides when they label the dialogs. In some dialogs, the systems provide the bus information that cannot exactly meet the user’s request, for example, wrong bus arrival time though correct bus route. The experts regard such dialogs as failure, while workers often consider them as success. On the other hand, workers seem to focus on the user’s final intent. In some cases where the user changes his mind and triggers a new query after the system provides acceptable information, the experts tag them as success, while workers often label them as failure since they feel
that the system does not accomplish the user’s final goal. Although the classification of task completion is more objective than other ratings (like user satisfaction), there are still some differences of opinion which may cause inconsistency.

1.7 Conclusions and Future Work

This chapter presented our strategy for the use of crowdsourcing in collection of user judgments on spoken dialog systems through crowdsourcing. We describe a design methodology for two types of tasks - the first for rapid, efficient collection of ratings of a large number of dialogs and the second for assessment of the reliability of these ratings through interannotator agreement among Workers. These tasks were designed specifically to overcome several challenges associated with crowdsourcing of SDS evaluation: creating a representative sample of dialogs for evaluation, providing sufficiently clear, simple tasks for Workers, and establishing a framework for semi-automatic validation of crowdsourced results. A set of approval rules ensured the quality of ratings from crowdsourcing.

Compared with the traditional method of inviting subjects to fill out a questionnaire after interaction, the results we achieved show that the crowdsourcing method is more efficient, flexible and inexpensive, and could access a more statistically representative population. At the same time, the quality of ratings can also be controlled. Reliable ratings for 8,394 dialogs rated by around 700 online workers are approved. Approval rates for each dialog category, i.e., TaskComplete, TaskIncomplete and OutofScope, are 79.59%, 65.23% and 90.65% respectively. Reasonable consistency between the manual crowdsourced ratings and the automatic dialog categorization in terms of task success is an indicator of the reliability of the approved ratings from crowdsourcing. The moderate level of interannotator agreement among workers for ratings in task completion partially verifies the reliability of such workers.

This chapter has also provided support for the use of crowdsourcing for evaluation of SDSs by investigating the agreement between workers and experts through two case studies. Experimental results showed a high level of interannotator agreement (around 0.7) between experts and workers in terms of task completion when we compared their annotations on the Let’s Go! dialog corpus. There was also a high degree of consistency between expert and non-expert labels of task success for the dialogs from the four SDC systems.

Acknowledgments

This work was done as part of a project related to the Spoken Dialog Challenge in 2010 (http://dialrc.org/sdc), which was organized by Professor Maxine Eskenazi and Professor Alan Black of the CMU Dialog Research Center. This work was conducted when Zhaojun Yang was a Masters student and Professor Gina-Anne Levow was a Visiting Scholar at The Chinese University of Hong Kong. The project team also included Professor Irwin King, Baichuan Li and Yi Zhu from The Chinese University of Hong Kong. The work is partially supported by a grant from the HKSAR Government Research Grants Council (Project No. 415609).
References

Baptist L and Seneff S 2000 Genesis-ii: A versatile system for language generation in conversational system applications Proceedings of ICSLP.


DePalma N, Chernova S and Breazeal C 2011 Leveraging online virtual agents to crowdsource human-robot interaction Proceedings of CHI Workshop on Crowdsourcing and Human Computation.


Engelbrecht K and Möller S 2007 Pragmatic usage of linear regression models for the prediction of user judgments Proc. of SIGdial Workshop on Discourse and Dialogue.


McTear M 2002 Spoken dialogue technology: enabling the conversational user interface. ACM Computing Surveys (CSUR) 34(1), 169.


Meng H, Lam W and Wai C 1999 To believe is to understand Proc. of European Conference on Speech Communication and Technology.


Möller S 2005 Parameters for quantifying the interaction with spoken dialogue telephone services Proc. of SIGdial Workshop on Discourse and Dialogue, pp. 166–177.

Möller S and Skowronek J 2003 Quantifying the impact of system characteristics on perceived quality dimensions of a spoken dialogue service Eighth European Conference on Speech Communication and Technology.


International Telecommunication Union, Geneva.

Score studies Proceeding of ICASSP 2011.

61–86.


Shoukri M 2004 Measures of interobserver agreement. CRC Press. 

Snow R, O’Connor B, Jurafsky D and Ng A 2008 Cheap and fast—but is it good?: evaluating non-expert annotations 
for natural language tasks Proc. of the Conference on Empirical Methods in Natural Language Processing, 
pp. 254–263.


Walker M, Kamm C and Litman D 2000 Towards developing general models of usability with paradise. Natural 

Walker M, Litman D, Kamm C and Abella A 1997 PARADISE: a framework for evaluating spoken dialogue agents 
Proc. of Association for Computational Linguistics (ACL). 

Walker M, Passonneau R and Boland J 2001 Quantitative and qualitative evaluation of DARPA Communicator 

Walker MA, Litman DJ, Kamm CA and Abella A 1998 Evaluating spoken dialogue agents with PARADISE: Two 

Ward W and Issar S 1994 Recent improvements in the CMU spoken language understanding system Proc. of the 

Wolters M, Isaac K and Renals S 2010 Evaluating speech synthesis intelligibility using Amazon Mechanical Turk 
Proceedings 7th Speech Synthesis Workshop (SSW7). 

Wu Z, Meng H, Ning H and Tse S 2006 A corpus-based approach for cooperative response generation in a dialog 

Yang Z, Li B, Zhu Y, King I, Levow G and Meng H 2010 Collection of User Judgments on Spoken Dialog System 
with Crowdsourcing Proc. of IEEE Workshop on Spoken Language Technology (SLT).
<table>
<thead>
<tr>
<th>Task</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task for dialog evaluation</td>
<td>12</td>
</tr>
<tr>
<td>Task for interannotator agreement</td>
<td>12</td>
</tr>
<tr>
<td>Workers</td>
<td>14</td>
</tr>
<tr>
<td>Workers Comments</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Mechanical Turk</td>
</tr>
<tr>
<td>Approval rules</td>
</tr>
<tr>
<td>Collection of user judgments</td>
</tr>
<tr>
<td>Consistency between automatic and manual ratings</td>
</tr>
<tr>
<td>Consistency between Workers and experts</td>
</tr>
<tr>
<td>Crowdsourcing</td>
</tr>
<tr>
<td>Dialog classification</td>
</tr>
<tr>
<td>Dialog systems</td>
</tr>
<tr>
<td>Interannotator agreement among Workers</td>
</tr>
<tr>
<td>Interannotator agreement between Workers and experts</td>
</tr>
<tr>
<td>ITU recommendations</td>
</tr>
<tr>
<td>Let’s Go</td>
</tr>
<tr>
<td>Mean Opinion Scores</td>
</tr>
<tr>
<td>Metrics interaction</td>
</tr>
<tr>
<td>Metrics subjective</td>
</tr>
<tr>
<td>PARADISE</td>
</tr>
<tr>
<td>SASSI</td>
</tr>
<tr>
<td>Speech synthesis assessment</td>
</tr>
<tr>
<td>Speech synthesis assessment of Spoken Dialog Challenge 2010</td>
</tr>
<tr>
<td>Spoken Dialog Challenge 2010</td>
</tr>
<tr>
<td>Spoken dialog system</td>
</tr>
<tr>
<td>Spoken dialog system evaluation component-based</td>
</tr>
<tr>
<td>Spoken dialog system evaluation holistic</td>
</tr>
<tr>
<td>Task</td>
</tr>
</tbody>
</table>