# SPOKEN DIALOG CHALLENGE 2010

*Alan W Black, Susanne Burger, Brian Langner, Gabriel Parent and Maxine Eskenazi*

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA
{awb,sburger,blangner,gparent, max}@cs.cmu.edu

## ABSTRACT

This paper describes the 2010 Spoken Dialog Challenge, a multi-site challenge to help investigate different spoken dialog system and evaluation techniques. The aim of the Challenge is to bring together multiple implementations of the same dialog task and deploy them in uncontrolled real user conditions and then make the results available for common evaluation techniques. This paper gives an overview of the Challenge itself and the task, and presents the results for the "controlled task" part of the evaluation. The paper also discusses the infrastructure and organizational issues encountered and the solutions that made this challenge possible.

***Index Terms***— spoken dialog, dialog evaluation

## 1. INTRODUCTION

One of the difficult aspects of research and development of complex systems is being able to realistically evaluate algorithms and techniques. Evaluation means comparison to others' results. The fact that usage patterns of each system are very different from every other one makes this task difficult. In the field of Spoken Dialog Systems (SDS) it has always been a research question in itself to compare systems and techniques across different deployments and numerous variables such as: the task, the user populations the platform and the evaluation metrics themselves. The NSF-funded DialRC (dialrc.org) project at CMU gives us the chance to bring the community together for a Challenge on one common task that enables realistic comparisons of systems and evaluation techniques.

The most similar previous attempt is probably the DARPA Communicator project of about 10 years ago, where a number of US sites built telephone-based SDS for flight information [1]. Their evaluation used a common set of callers for all systems. This project brought together many different groups, all centered in the US, who helped determine the spoken dialog techniques that were helpful and allowed better comparison amongst them. However, despite the increasing maturity of SDS in the past decade, nothing of this scale has been attempted since then.

One specific aspect of SDS that we wished to address in the 2010 Challenge is that of user type. It has been shown that paid callers are very different from 'real" callers [2]. The standard, perhaps apocryphal, example is a caller who asks to fly to "San Diego" but is willing to accept a flight to "San Jose" even though that is more than 700km from where they are supposed to go. In choosing the common task for the Challenge, it was critical that the task be applied to real users, people who want to use the system for the information that it provides and not because they are paid, or "just" wish to improve the state of SDS. Apart from the difference in their performance, real users of a system, if it is useful to them, can furnish a much larger amount of data in less time at a much lower cost both in human effort to get them and in monetary resources.

The Spoken Dialog Challenge was initiated in 2009 through discussions with the community of those who develop and evaluate SDS. It was codified in the public 2010 Challenge announcement at SigDIAL 2009 [3].

## 2. SDC 2010 TASK: LET'S GO BUS SCHEDULE

The Let's Go Bus information system [4] has been used nightly by real callers in Pittsburgh since March 2005 and has answered over 100,000 calls. The system is based on the CMU Olympus Spoken Dialog System [5] and gives bus times based on the local Port Authority schedules for the East End of Pittsburgh. The dialogs may be considered by some to be rather simple, yet they still manifest sufficient complexity to provide an important research platform, where new research ideas can be tested with real users. The system answers the phone on evenings and weekends and deals with all of the real world issues and goals, from new users who have never spoken to machines before to drunks calling in the middle of the night. The Let's Go system has an estimated task success rate of just under 75%.

The logs from the 100K calls are available to the research community, making this one of the few (and possibly the only) publicly available SDSs (and corresponding dataset) that has a significant number of logs from calls from real users.

The Let's Go task was chosen as the 2010 Challenge task because it had a system that already existed and a substantial set of collected dialogs. There was also a reference implementation that could be distributed to participants if they wished to start from the CMU systems, and most importantly there was a real user base that could be used to test the competing systems with a significant amount of calls.

The Let's Go system software is based on the CMU Olympus Dialog system. It consist of a number of modules, including the Pocket Sphinx recognizer, Phoenix parser, the Ravenclaw dialog manager, Rosetta generation system and Flite synthesizer. The whole system runs on a reasonably powered Windows XP machine. Detailed installation instructions are available, but in our experience installation of the whole system at present, still requires significant experience in software management.

In the initial stage the potential challenge participants were educated about the task and the existing system. Due to the interest from groups over several continents and in order to minimize time and travel costs, this was done through a series of webcasts.

The next stage was to distribute the 700GB of dialog logs to participants. At this point there were 10 groups who requested the data.

Although we have collected five years of data from Let's Go, only about five months of it had been transcribed as of the beginning of the Challenge. We had distributed those transcriptions along with the rest of the data with automatically applied labels and corresponding logfiles. However, we felt that it was essential to transcribe much more data both for training and for the simulated user thread. The monetary burden of expert labeling is formidable for this amount of data, as is the amount of time it would take to complete this task. Therefore we decided to use Amazon Mechanical Turk (AMT) to label one year of Let's Go data (2008-2009). We constructed the HITs so that they included gold standard controls. We also implemented a two-pass approach to ensure the quality of the output [6]. The data has been analyzed and distributed and is now publicly available.

## 3. THE CHALLENGE TASKS

The Challenge initially consisted of two commonly agreed upon thrusts with a third one added after several groups showed interest in it:

1. Build a telephone-based SDS that gives bus schedule information for the East End of Pittsburgh.
2. Devise an evaluation technique to compare multiple systems on the same task.
3. Build a simulated user for the system(s).

It is not a trivial task to build a working, robust, telephone dialog system addressing 8 bus routes, and 15,000 bus stops in a short amount of time. Thus it is rewarding to note that a number of groups rose to the Challenge. And a reflection of the great advancements in rapid dialog building techniques is that three groups were able to produce fully working systems in three months' time. Although it may not be evident from their own descriptions of their work, the task of building the systems did not always follow its originally intended paths this year.

Despite disposing of many sample dialogs, and a reference system, creating a working system using an unfamiliar dialog architecture demands a significant amount of work. Unlike other speech challenges, an SDS contains many different components that each benefit from careful tuning after substantial training, making the work for this challenge more complex. And to complicate matters, the participants had to deal with real world concerns such as updating the bus schedule timetables and addressing major route changes to some of the bus lines by the local Port Authority in the middle of the Challenge (the Port Authority changes schedules four times a year and only releases the new schedule a few days before it is to go into effect; it is also streamlining its operations and so each new schedule comes with changes to the bus lines).

## 4. DEPLOYMENT AND CORE EVALUATION

We chose two stages of deployment. The first stage was termed the "**control test**". In this part, the four systems (the three competitors and the CMU reference system) were made available (could be called from a phone number that was distributed). Each group was asked to provide their own set of callers who each called every one of the systems. This control was intended to both check the robustness of the systems and provide a set of results that could be compared to the real world deployed system. For many researchers control tests are the only type of test available (this is also the case for all systems when they are first constructed). Thus we wanted to understand how performance in a controlled situation correlates with real user performance.

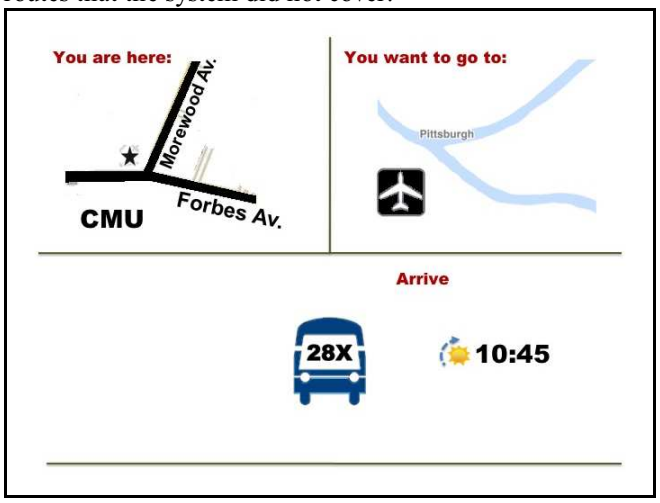
The second "**deployment**" stage involved taking the three of the four systems that were considered robust enough to deal with real users "live", providing real service to the Pittsburgh community. The fourth system was considered not stable enough to go forward, although it is yet to be determined whether this is due to the telephone connection or to the performance of the system itself.

## 5. CONTROL TESTS

### 5.1. Control Tests

The control tests were set up as a web page that would lead the subject through 8 bus scenarios. Each subject called 4 different systems twice each during the test. Due to the multinational aspect of the participants the numbers to be called were in different countries (and often different from the caller's own country). This may have caused some callers to be reluctant to pay calling charges but it was probably the only reasonable solution for this first Challenge.

The scenarios were presented symbolically through pictures and diagrams in order to minimize priming the subjects for specific words and phrases. We chose scenarios that reflected the call type distribution of the real callers of the Let's Go System: using scenarios that specified a bus number, departure place, departure time (or arrival time) and also scenarios that made requests for routes that the system did not cover.




Above is an example of a scenario picture: Scenario C2: From CMU/Morewood Ave at Forbes Ave to Pittsburgh airport, arrive at 10:45AM.

We were very much aware that the subject population for the control tests would mostly not be familiar with local Pittsburgh geography or the bus routes and took that into account when devising the web interface. We did try to select place names that would have a single standard pronunciation for the majority of English speakers – of course our calling population included many different English dialects including non-native dialects. 36% of all calls had speakers with detectable non-native English accents. In comparison, a subset of three months of daily Let's Go recording shows only 1.2% of non-English accents (estimated from a 1156 subset of Let's Go dialogs).

All control test dialogs were manually transcribed. The transcribed dialogs were matched to the log files that each site provided. We logged the time when callers accessed the scenarios presented on the web page, and collected caller information and caller comments (each caller also entered schedule information that they received in the call, and/or other comments).

The web logs were then matched to those transcribed dialogs that had a matching system log file. At times, the time of web access and the call time had differences of between less than a minute and several minutes. This was due to the log access being later than the call or vice versa. Delays could have been caused by an intermediate call, or mismatch in times on different machines

Times were also adapted to time zone differences. Not all dialogs had matching web logs: there were web logs without matching dialog transcriptions, and transcribed dialogs that did not have a matching web log. Table 1 shows the number of web logs, the number of transcribed dialogs, and the number of dialogs that had a transcription, a matching system log file *and* a matching web log. Only these dialogs became part of the control test data set.

| | CMU | SYS2 | SYS3 | SYS4 |
|---|---|---|---|---|
| **Web logs** | 91 | 92 | 91 | 92 |
| **System logs** | 91 | 96 | 119 | 97 |
| **Matched logs** | 91 | 61 | 75 | 83 |

**Table 1**: Web vs System logs

The extra system logs were due to other calls made to the systems during the control tests, and the missing logs were most often due to failures of the control participants to connect to the target systems (due to either telephone network problems and problems with the systems themselves).

### 5.2 Control Test Metrics

We wish to underline the fact that the purpose of the Spoken Dialog Challenge is not to find a best system. It is to provide an opportunity to compare different systems on one common task. We can in fact achieve this goal better when the participating systems demonstrate different performance from one another. The following table gives the reader an overview of the performance of the four *control test* systems over a variety of metrics.

| | CMU | SYS2 | SYS3 | SYS4 |
|---|---|---|---|---|
| **audio_segs** | 1664 | 1072 | 771 | 1220 |
| **segs/dialog** | 18.29 | 17.57 | 10.28 | 14.70 |
| **max segs** | 59 | 59 | 80 | 84 |
| **min segs** | 3 | 5 | 2 | 1 |
| **empty%** | 7.21% | 28.26% | 8.95% | 6.72% |
| **wrd/seg-c** | 2.36 (2.26) | 1.09 (0.97) | 1.85 (1.29) | 1.72 (1.78) |
| **wrd/seg-a** | 2.87 (2.84) | 1.63 (1.62) | 2.73 (1.94) | 2.25 (1.78) |

**Table 2:** Dialog Segment Analysis

**audio_segs:** total number of audio segments
**segs/dialog:** average of audio segments per dialog
**max-segs:** maximum number of segments per dialog
**min-segs**: minimum number of segments per dialog
**empty%**: percentage of all audio segments per system which contain only silence or noise
**wrd/seg-c**: average (stddev) words per seg with concepts treated as single words (e.g. **forbes_and_craig**)
**wrd/seg-w**: average (stddev) words per audio segment: with concepts treated as single words (e.g. **forbes and craig**)

| | CMU | SYS2 | SYS3 | SYS4 |
|---|---|---|---|---|
| **noise** | 19.59% | 3.54% | 5.06% | 6.48% |
| **non_words** | 14.60% | 5.41% | 11.28% | 10.74% |
| **(re_)start** | 3.85% | 0.84% | 3.50% | 2.62% |
| **repeat** | 2.58% | 0.84% | 4.80% | 4.92% |
| **go back** | 0.18% | 0.19% | 0.13% | 2.70% |
| **next** | 3.06% | 1.31% | 4.54% | 4.51% |
| **previous** | 1.98% | 0.19% | 1.17% | 3.03% |
| **place:p** | 24.34% | 23.79% | 32.81% | 25.66% |
| **time:t** | 12.20% | 10.63% | 12.71% | 13.77% |
| **bus:b** | 2.46% | 0.37% | 0.78% | 4.92% |
| **polite** | 6.97% | 2.61% | 9.47% | 3.93% |

Table 3: Dialog Segment Content Analysis

Table 3 shows percentages of total of segments containing one or more of the following features;

**noise**: any noise not produced by caller's voice (background talking, technical noises)
**non_words**: caller's speech not resulting in understandable words (broken words, filled pauses, non-understandable words)
**command words**: expressions to direct the dialog: (re)/start/ (restart, let's start again, start a new query …) /repeat/, /go back/, /next/ and /previous/
**information requests**: actual input of departure and arrival location (**place:p**) time requests (**time:t**) and bus numbers (**bus:b**)
**polite**: expressions of politeness: greetings (hello, hi, good-day, goodbye, etc), excuses (sorry, pardon, excuse), please and thanking.

We also looked at two types of **success**:
**Any-Output**: the system gives some information: schedule information, a message about not covering a bus route or neighborhood, or a message that there is no bus at the requested time. Any of these forms of output is considered to be a success. Dialogs that do not show any of these forms of output are not considered successful. Note that in this form of successful dialog, the information may not actually provide an acceptable answer to the caller's request.
**Correct-Output**: the system gives some information: schedule information, a message about not covering a bus route or neighborhood, or a message that there is no bus at the requested time. However, only output messages which give an **acceptable** answer to the caller's request are considered to be successful outputs. Messages with **incorrect** departure and arrival stops or incorrect time are not successful.

Importantly, the difference between these two measures is that the first type can be easily calculated automatically, while the second requires human labeling. Also in a live system there are other calls that do not give any information since the caller hangs up, does not get what they want, or (sometimes) the bus they are waiting for arrives.

A successful call has at least one acceptable answer. An acceptable answer provides:

- schedule information for requested departure and arrival stops, requested time.
- a message that there is no coverage for the requested route or neighborhood
- a message that there is no bus for the requested time

We also allowed the following boundary cases:

- a departure or arrival stop that can be reached within 15 minutes by walking
- departure and arrival times within the range of an hour before or after the request.
- any bus numbers that serve the desired route.

An example of a successful call would be: The caller requested *From Forbes and Craig to the Waterfront, arriving by 9AM.* The system provides an answer*: 61D leaving FORBES AVENUE OPPOSITE MOREWOOD CARNEGIE MELLON at 8 30 a.m. It will arrive at WATERFRONT at TERMINUS at 8 58 a.m.* The given departure stop is only 6 minutes walking distance from Craig/Forbes, and, therefore, still considered acceptable.

| | CMU | SYS2 | SYS3 | SYS4 |
|---|---|---|---|---|
| **Any-Output** | 96.70% | 62.30% | 98.67% | 90.36% |
| **Acceptable** | 64.84% | 37.70% | 89.33% | 74.70% |
| **Incorrect** | 31.87% | 24.59% | 9.33% | 15.66% |
| **No output** | 3.30% | 37.70% | 1.33% | 9.64% |

Table 4: Success: Percentage of dialogs per system.

The measures used in Table 4 are:

**Any-Output**: any output is accepted;

**Acceptable**: dialogs with at least one acceptable information;

**Incorrect**: dialogs with only incorrect information;

**No output**: dialogs that do not provide any output message to the caller.

We observed the following for two systems:

**SYS2** had irregularities in the log files. For 14 dialogs, the system did not output the requested information to the callers, even if the log files actually showed that information was found. Since the caller never got the information, it could not be included in the success labeling.

**SYS3** also had irregularities in the log files. During transcription, 15 dialogs did not have an audio file for the first user turn. These turns were not transcribed. However, the system originally had heard the caller turn and had produced ASR and output for it. Success labeling included this information without transcription because it didn't interfere with the dialog.

At this stage we do not believe these difference are significant, as the total number of calls per system was (at most) under 100, and there were a number of infrastructure influences we find it prudent to be conservative on what we can infer from them.

However we believe we can attribute the longer number of turns to the CMU system being due to more explicit confirmation that used by the other systems. It also appears to be that the users use more words when taking to the CMU system than the others.

## 6. LIVE TESTS

The systems that appeared the most reliable, the **CMU** benchmark system, **SYS3** and **SYS4** were chosen to go on to the next stage. They were judged to be reliable due to their robust performance during the control tests. CMU's benchmark system was included in this group. However since the CMU system had already serviced many calls on the live telephone number, we organized a schedule for the two other systems that gave them more live time than the CMU system, particularly on weekends when there are more calls.

A non-trivial aspect was how to provide access to an SDS in a way that, when a Pittsburgh caller calls the Port Authority phone number, they are appropriately (and quickly) forwarded to the target system which could be located in another country. We would like to thank AT&T for providing a method for a US telephone number to be redirected to an international number. This enabled all of the systems to be called through a simple (but long) series of telephone redirections. Each day the appropriate system was put online by changing the redirection.

For the period of mid-July to mid-August 2010 (once the selected system had again updated its bus schedule databases) the people of Pittsburgh were provided with a group of differently designed telephone dialog systems, a different one each evening.

As of the writing of this paper, we are still interpreting those results and so they are not included here. However our initial observations show that the systems all have performed well with no significantly different success rates compared to the reference CMU system.

## 6. CHALLENGE LESSONS LEARNED

Running a large challenge, across a large number of groups, especially when there is no funding related to the work other than CMU's NSF grant that can support infrastructure only, is a difficult task. Although we already have experience in running large spoken language challenges, this Challenge gave rise to a variety of issues both expected and unexpected.

Although we could provide a substantial amount of data, its format was not always in the form that others expected. We do include substantial varied logs and methods to parse these logs but of course when others used the output, they still needed to take some time to format it for their needs. Even though we provided distribution through 1TB drives, it still took almost a day to copy the data each time. Thus distribution of all the data to the sites ended up taking longer than we intended. Text-only logs (around 25GB) were distributed online but networks are not yet fast enough to distribute 700GB in a reasonable time around the world.

Although we provided a full working reference system to groups who intended to use that system (and replace one or more of its components), we discovered that this is not always as easy as expected. One group requested n-best lists from our recognizer to use to train their system. Although our recognizer can produce n-best lists we have not done that in Let's Go. An attempt to regenerate an n-best list was made, but only in that group's desired format.

We deliberately included a control test, but, perhaps ironically, we were hit with the problem all lab-based tests have: finding callers. We took longer to get a sufficient number of callers and had to extend our intended time so that we could chase up additional callers. We were also aware that our control test population was very different from real users of the bus schedule system. Most were *not* familiar with Pittsburgh, most were *very* familiar with SDSs, and they typically *tried harder* to complete the task than our real users do. It is interesting to include such a test in a challenge (as it allows comparison with real users), but we underestimated the difficulty of getting callers. In future challenges we will have to spend more

time to collect such users, ahead of the time when they are expected to make the calls, and find some additional ways to incentivize them. We believe that we still need to get callers from all of the sites participating in the Challenge and that there should be an approximately equal number from each one.

Providing telephone infrastructure was also an unknown, but this turned out not to be a problem. Many people believe that voice over IP solutions are readily available but that connecting these reliably across multiple telephone networks is not trivial. For our control tests we asked participants not to use Skype (or similar) systems because of the different encoding schemes and additional delays. Some still did. The live Let's Go system rarely receives VOIP calls, so systems would not have any advantage to specifically tune towards this.

We do have resources for labeling data, but it takes time to produce hand labeling, even when labeling occurs at the same time that other tests are being performed.

We started with a clear timetable for education, data distribution, development and evaluation. But as is often the case in the first trial in a major undertaking, the deadlines slipped. Some causes were due to external circumstances, but we also now have a more realistic estimate of the amount of time to carry out each of the tasks.

## 8. CONCLUSIONS

It is our desire to support a Spoken Dialog Challenge (SDC) as an annual event, but to do so will require a tighter timetable, particularly in getting the participants to be directly involved earlier in the process. It also necessitates real users in a real application and much data.

If we aim for presentation of results at some event in the fall of a Challenge year, it is clear that testing of the systems has to happen much earlier that year than in the spring, in order for labeling and data analysis to take place in time for paper deadlines. Thus system construction has to be moved to the fall of the previous year in order to meet the deadlines.

We believe that the Spoken Dialog Challenge is a success but that it is only the start of series of challenges that should continue to be shaped by the dialog community in order to aid advancement of spoken dialog systems.

## 9. ACKNOWLEDGEMENTS

This work was in part supported by the US National Science foundation under the project "Dialogue Research Center". We also wish to thank the participants of the challenge who made it possible. We would also like to specifically thanks Jason Williams (AT&T Research) for providing support for in our telephone infrastructure.

## 10. REFERENCES

[1] Walker, M., Passonneau, R., and Boland, J. "Quantitative and qualitative evaluation of DARPA Communicator spoken dialogue systems" 39th ACL, pp 515-222, Toulouse, France, 2001.

[2] Ai, H., Raux, A., Bohus, D., Eskenazi, M., and Litman, D. (2007) "Comparing Spoken Dialog Corpora Collected with Recruited Subjects versus Real Users", in SIGDial Workshop on Discourse and Dialogue, Antwerp, Belgium.

[3] Black, A. and Eskenazi, M. "The Spoken Dialog Challenge", SIGDial 2009, Queen Mary University, London, 2009.

[4] Raux, A., Bohus, D., Langner, B., Black, A., and Eskenazi, M. (2006) "Doing Research on a Deployed Spoken Dialogue System: One Year of Let's Go! Experience", Interspeech 2006 - ICSLP, Pittsburgh, PA.

[5] Olympus II Dialog System Framework http://wiki.speech.cs.cmu.edu/olympus/index.php/Olympus

[6] Parent, G. and Eskenazi, M. "Toward better crowdsourced transcription: transcription of a year of the Let's Go Bus Information System" submitted to SLT 2010.