Spoken Dialog Challenge 2010: Comparison of Live and Control Test Results

Alan W Black[1], Susanne Burger[1], Alistair Conkie[4], Helen Hastie[2], Simon Keizer[3], Oliver Lemon[2], Nicolas Merigaud[2], Gabriel Parent[1], Gabriel Schubiner[1], Blaise Thomson[3], Jason D. Williams[4], Kai Yu[3], Steve Young[3] and Maxine Eskenazi[1]

[1]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA
[2]Dept of Mathematical and Computer Science, Heriot-Watt University, Edinburgh, UK
[3]Engineering Department, Cambridge University, Cambridge, UK
[4]AT&T Labs – Research, Florham Park, NJ, USA
awb@cs.cmu.edu

Abstract

The Spoken Dialog Challenge 2010 was an exercise to investigate how different spoken dialog systems perform on the same task. The existing Let's Go Pittsburgh Bus Information System was used as a task and four teams provided systems that were first tested in controlled conditions with speech researchers as users. The three most stable systems were then deployed to real callers. This paper presents the results of the live tests, and compares them with the control test results. Results show considerable variation both between systems and between the control and live tests. Interestingly, relatively high task completion for controlled tests did not always predict relatively high task completion for live tests. Moreover, even though the systems were quite different in their designs, we saw very similar correlations between word error rate and task completion for all the systems. The dialog data collected is available to the research community.

1 Background

The goal of the Spoken Dialog Challenge (SDC) is to investigate how different dialog systems perform on a similar task. It is designed as a regularly recurring challenge. The first one took place in 2010. SDC participants were to provide one or more of three things: a system; a simulated user, and/or an evaluation metric. The task chosen for the first SDC was one that already had a large number of real callers. This had several advantages. First, there was a system that had been used by many callers. Second, there was a substantial dataset that participants could use to train their systems. Finally, there were real callers, rather than only lab testers. Past work has found systems which appear to perform well in lab tests do not always perform well when deployed to real callers, in part because real callers behave differently than lab testers, and usage conditions can be considerably different [Raux et al 2005, Ai et al 2008]. Deploying systems to real users is an important trait of the Spoken Dialog Challenge.

The CMU Let's Go Bus Information system [Raux et al 2006] provides bus schedule information for the general population of Pittsburgh. It is directly connected to the local Port Authority, whose evening calls for bus information are redirected to the automated system. The system has been running since March 2005 and has served over 130K calls.

The software and the previous years of dialog data were released to participants of the challenge to allow them to construct their own systems. A number of sites started the challenge, and four sites successfully built systems, including the original CMU system.

An important aspect of the challenge is that the quality of service to the end users (people in Pittsburgh) had to be maintained and thus an initial robustness and quality test was carried out on contributed systems. This control test provided scenarios over a web interface and required researchers from the participating sites to call each of the systems. The results of this control test were published in [Black et al. 2010] and by the individual participants [Williams et al. 2010, Thomson et al. 2010, Hastie et al, 2010] and they are repro-

Proceedings of the SIGDIAL 2011: the 12th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 2–7,
Portland, Oregon, June 17-18, 2011.

duced below to give the reader a comparison with the later live tests.

Important distinctions between the control test callers and the live test callers were that the control test callers were primarily spoken dialog researchers from around the world. Although they were usually calling from more controlled acoustic conditions, most were not knowledgeable about Pittsburgh geography.

As mentioned above, four systems took part in the SDC. Following the practice of other challenges, we will not explicitly identify the sites where these systems were developed. We simply refer to them as SYS1-4 in the results. We will, however, state that one of the systems is the system that has been running for this task for several years. The architectures of the systems cover a number of different techniques for building spoken dialog systems, including agenda based systems, VoiceXML and statistical techniques.

2 Conditions of Control and Live tests

For this task, the caller needs to provide the departure stop, the arrival stop and the time of departure or arrival in order for the system to be able to perform a lookup in the schedule database. The route number can also be provided and used in the lookup, but it is not necessary. The present live system covers the East End of Pittsburgh. Although the Port Authority message states that other areas are not covered, callers may still ask for routes that are not in the East End; in this case, the live system must say it doesn't have information available. Some events that affect the length of the dialog include whether the system uses implicit or explicit confirmation or some combination of both, whether the system has an open-ended first turn or a directed one, and whether it deals with requests for the previous and/or following bus (this latter should have been present in all of the systems).

Just before the SDC started, the Port Authority had removed some of its bus routes. The systems were required to be capable of informing the caller that the route had been canceled, and then giving them a suitable alternative.

SDC systems answer live calls when the Port Authority call center is closed in the evening and early morning. There are quite different types and volumes of calls over the different days of the week. Weekend days typically have more calls, in part because the call center is open fewer hours on weekends. Figure 1 shows a histogram of average calls per hour for the evening and the early morning of each day of the week.

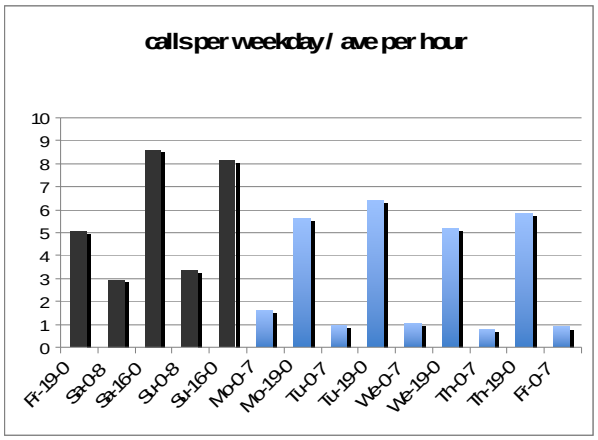

Figure 1: average number of calls per hour on weekends (dark bars) and weekdays. Listed are names of days and times before and after midnight when callers called the system.

The control tests were set up through a simple web interface that presented 8 different scenarios to callers. Callers were given a phone number to call; each caller spoke to each of the 4 different systems twice. A typical scenario was presented with few words, mainly relying on graphics in order to avoid influencing the caller's choice of vocabulary. An example is shown in Figure 2.

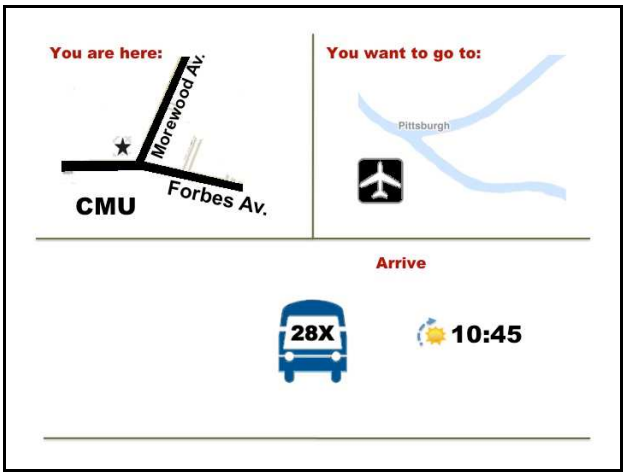

Figure 2: Typical scenario for the control tests. This example requests that the user find a bus from the corner of Forbes and Morewood (near CMU) to the airport, using bus route 28X, arriving by 10:45 AM.

3 Control Test Results

The logs from the four systems were labeled for task success by hand. A call is successful if any of the following outputs are correctly issued:

- Bus schedule for the requested departure and arrival stops for the stated bus number (if given).
- A statement that there is no bus available for that route.
- A statement that there is no scheduled bus at that time.

We additionally allowed the following boundary cases:

- A departure/arrival stop within 15 minutes walk.
- Departure/arrival times within one hour of requested time.
- An alternate bus number that serves the requested route.

In the control tests, SYS2 had system connection issues that caused a number of calls to fail to connect, as well as a poorer task completion. It was not included in the live tests. It should be pointed out that SYS2 was developed by a single graduate student as a class project while the other systems were developed by teams of researchers. The results of the Control Tests are shown in Table 1 and are discussed further below.

	SYS1	**SYS2**	**SYS3**	**SYS4**
Total Calls	91	61	75	83
no_ info	3.3%	37.7%	1.3%	9.6%
donthave	17.6%	24.6%	14.7%	9.6%
donthave_corr	*68.8%*	*33.3%*	*100.0%*	*100.0%*
donthave_incorr	*31.3%*	*66.7%*	*0.0%*	*0.0%*
pos_out	79.1%	37.7%	84.0%	80.7%
pos_out_corr	*66.7%*	*78.3%*	*88.9%*	*80.6%*
pos_out_incorr	*33.3%*	*21.7%*	*11.1%*	*19.4%*

Table 1. Results of hand analysis of the four systems in the **control test**

The three major classes of system response are as follows. **no_info**: this occurs when the system gives neither a specific time nor a valid excuse (bus not covered, or none at that time). **no_info** calls can be treated as errors (even though there maybe be valid reasons such as the caller hangs up because the bus they are waiting for arrives). **donthave**: identifies calls that state the requested bus is not covered by the system or that there is no bus at the requested time. **pos_out**: identifies calls where a specific time schedule is given. Both **donthave** and **pos_out** calls may be correct or erroneous (e.g the given information is not for the requested bus, the departure stop is wrong, etc).

4 Live Tests Results

In the live tests the actual Pittsburgh callers had access to three systems: SYS1, SYS3, and SYS4. Although engineering issues may not always be seen to be as relevant as scientific results, it is important to acknowledge several issues that had to be overcome in order to run the live tests.

Since the Pittsburgh Bus Information System is a real system, it is regularly updated with new schedules from the Port Authority. This happens about every three months and sometimes includes changes in bus routes as well as times and stops. The SDC participants were given these updates and were allowed the time to make the changes to their systems. Making things more difficult is the fact that the Port Authority often only releases the schedules a few days ahead of the change. Another concern was that the live tests be run within one schedule period so that the change in schedule would not affect the results.

The second engineering issue concerned telephony connectivity. There had to be a way to transfer calls from the Port Authority to the participating systems (that were run at the participating sites, not at CMU) without slowing down or perturbing service to the callers. This was achieved by an elaborate set of call-forwarding mechanisms that performed very reliably. However, since one system was in Europe, connections to it were sometimes not as reliable as to the US-based systems.

	SYS1	**SYS3**	**SYS4**
Total Calls	678	451	742
Non-empty calls	633	430	670
no_ info	18.5%	14.0%	11.0%
donthave	26.4%	30.0%	17.6%
donthave_corr	*47.3%*	*40.3%*	*37.3%*
donthave_incorr	*52.7%*	*59.7%*	*62.7%*
pos_out	55.1%	56.0%	71.3%
pos_out_corr	*86.8%*	*93.8%*	*91.6%*
pos_out_incorr	*13.2%*	*6.2%*	*8.4%*

Table 2. Results of hand analysis of the three systems in the **live tests**. Row labels are the same as in Table 1.

We ran each of the three systems for multiple two day periods over July and August 2010. This design gave each system an equal distribution of weekdays and weekends, and also ensured that repeat-callers within the same day experienced the same system.

One of the participating systems (SYS4) could support simultaneous calls, but the other two could not and the caller would receive a busy signal if the system was already in use. This, however, did not happen very often.

Results of hand analysis of real calls are shown in Table 4 alongside the results for the Control Test for easy comparison. In the live tests we had an additional category of call types – empty calls (0-turn calls) – which are calls where there are no user turns, for example because the caller hung up or was disconnected before saying anything. Each system had 14 days of calls and external daily factors may change the number of calls. We do suspect that telephony issues may have prevented some calls from getting through to SYS3 on some occasions.

Table 3 provides call duration information for each of the systems in both the control and live tests.

	Length (s)	**Turns/call**	**Words/turn**
SYS1 control	155	18.29	2.87 (2.84)
SYS1 live	111	16.24	2.15 (1.03)
SYS2 control	147	17.57	1.63 (1.62)
SYS3 control	96	10.28	2.73 (1.94)
SYS3 live	80	9.56	2.22 (1.14)
SYS4 control	154	14.70	2.25 (1.78)
SYS4 live	126	11.00	1.63 (0.77)

Table 3: For live tests, average length of each call, average number of turns per call, and average number of words per turn (numbers in brackets are standard deviations).

Each of the systems used a different speech recognizer. In order to understand the impact of word error rate on the results, all the data were hand transcribed to provide orthographic transcriptions of each user turn. Summary word error statistics are shown in Table 4. However, summary statistics do not show the correlation between word error rate and dialogue success. To achieve this, following Thomson et al (2010), we computed a logistic regression of success against word error rate (WER) for each of the systems. Figure 3 shows the regressions for the Control Tests and Figure 4 for the Live Tests.

	SYS1	**SYS3**	**SYS4**
Control	38.4	27.9	27.5
Live	43.8	42.5	35.7

Table 4: Average dialogue word error rate (WER).

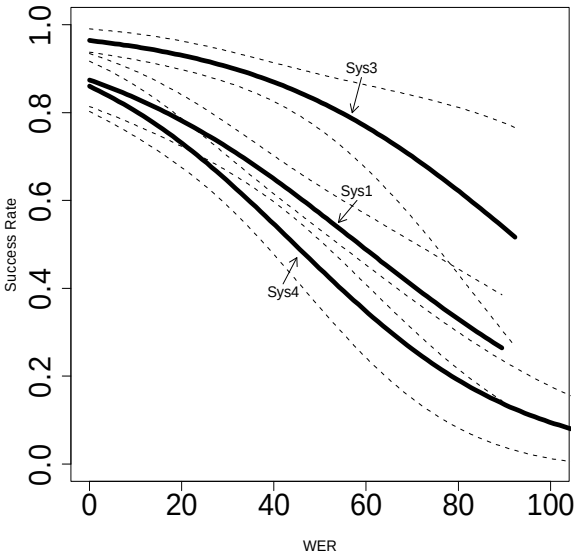

Figure 3: Logistic regression of control test success vs WER for the three fully tested systems

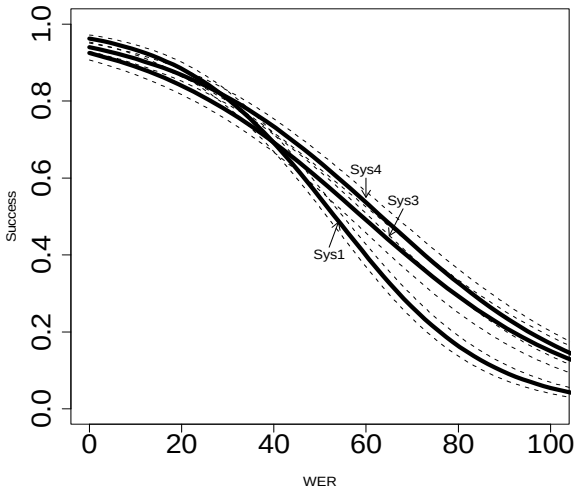

Figure 4: Logistic regression of live success vs WER for the three fully tested systems

In order to compare the control and live tests, we can calculate task completion as the percentage of calls that gave a correct result. We include only non-empty calls (excluding 0-turn calls), and treat all no_info calls as being incorrect, even though some may be due to extraneous reasons such as the bus turning up (Table 5).

	SYS1	**SYS3**	**SYS4**
Control	64.9% (5.0%)	89.4% (3.6%)	74.6% (4.8%)
Live	60.3% (1.9%)	64.6% (2.3%)	71.9% (1.7%)

Table 5: Live and control test task completion (std. err).

5 Discussion

All systems had lower WER and higher task completion in the controlled test vs. the live test. This agrees with past work [Raux et al 2005, Ai et al 2008], and underscores the challenges of deploying real-world systems.

For all systems, dialogs with controlled subjects were longer than with live callers – both in terms of length and number of turns. In addition, for all systems, live callers used shorter utterances than controlled subjects. Controlled subjects may be more patient than live callers, or perhaps live callers were more likely to abandon calls in the face of higher recognition error rates.

Some interesting differences between the systems are evident in the live tests. Looking at dialog durations, SYS3 used confirmations least often, and yielded the fastest dialogs (80s/call). SYS1 made extensive use of confirmations, yielding the most turns of any system and slightly longer dialogs (111s/call). SYS4 was the most system-directed, always collecting information one element at a time. As a result it was the slowest of the systems (126s/call), but because it often used implicit confirmation instead of explicit confirmation, it had fewer turns/call than SYS1.

For task completion, SYS3 performed best in the controlled trials, with SYS1 worst and SYS4 in between. However in the live test, SYS4 performed best, with SYS3 and SYS1 similar and worse. It was surprising that task completion for SYS3 was the highest for the controlled tests yet among the lowest for the live tests. Investigating this, we found that much of the variability in task completion for the live tests appears to be due to WER. In the control tests SYS3 and SYS4 had similar error rates but the success rate of SYS3 was higher. The regression in Figure 3 shows this clearly. In the live tests SYS3 had a significantly higher word error rate and average success rate was much lower than in SYS4.

It is interesting to speculate on why the recognition rates for SYS3 and SYS4 were different in the live tests, but were comparable in the control tests. In a spoken dialogue system the architecture has a considerable impact on the measured word error rate. Not only will the language model and use of dialogue context be different, but the dialogue design and form of system prompts will influence the form and content of user inputs. Thus, word error rates do not just depend on the quality of the acoustic models – they depend on the whole system design. As noted above, SYS4 was more system-directed than SYS3 and this probably contributed to the comparatively better ASR performance with live users. In the control tests, the behavior of users (research lab workers) may have been less dependent on the manner in which users were prompted for information by the system. Overall, of course, it is user satisfaction and task success which matter.

6 Corpus Availability and Evaluation

The SDC2010 database of all logs from all systems including audio plus hand transcribed utterances, and hand defined success values is released through CMU's Dialog Research Center (http://dialrc.org).

One of the core goals of the Spoken Dialog Challenge is to not only create an opportunity for researchers to test their systems on a common platform with real users, but also create common data sets for testing evaluation metrics. Although some work has been done on this for the control test data (e.g. [Zhu et al 2010]), we expect further evaluation techniques will be applied to these data.

One particular issue which arose during this evaluation concerned the difficulty of defining precisely what constitutes task success. A precise definition is important to developers, especially if reinforcement style learning is being used to optimize the success. In an information seeking task of the type described here, task success is straightforward when the user's requirements can be satisfied but more difficult if some form of constraint relaxation is required. For example, if the user

asks if there is a bus from the current location to the airport – the answer "No." may be strictly correct but not necessarily helpful. Should this dialogue be scored as successful or not? The answer "No, but there is a stop two blocks away where you can take the number 28X bus direct to the airport." is clearly more useful to the user. Should success therefore be a numeric measure rather than a binary decision? And if a measure, how can it be precisely defined? A second and related issue is the need for evaluation algorithms which determine task success automatically. Without these, system optimization will remain an art rather than a science.

7 Conclusions

This paper has described the first attempt at an exercise to investigate how different spoken dialog systems perform on the same task. The existing Let's Go Pittsburgh Bus Information System was used as a task and four teams provided systems that were first tested in controlled conditions with speech researchers as users. The three most stable systems were then deployed "live" with real callers. Results show considerable variation both between systems and between the control and live tests. Interestingly, relatively high task completion for controlled tests did not always predict relatively high task completion for live tests. This confirms the importance of testing on live callers, not just usability subjects.

The general organization and framework of the evaluation worked well. The ability to route audio telephone calls to anywhere in the world using voice over IP protocols was critical to the success of the challenge since it provides a way for individual research labs to test their in-house systems without the need to port them to a central coordinating site.

Finally, the critical role of precise evaluation metrics was noted and the need for automatic tools to compute them. Developers need these at an early stage in the cycle to ensure that when systems are subsequently evaluated, the results and system behaviors can be properly compared.

Acknowledgments

Thanks to AT&T Research for providing telephony support for transporting telephone calls during the live tests. This work was in part supported by the US National Science foundation under the project "Dialogue Research Center".

References

Ai, H., Raux, A., Bohus, D., Eskenzai, M., and Litman, D. (2008) "Comparing spoken dialog corpora collected with recruited subjects versus real users", Proc SIGDial, Columbus, Ohio, USA.

Black, A., Burger, S., Langner, B., Parent, G., and Eskenazi, M. (2010) "Spoken Dialog Challenge 2010", SLT 2010, Berkeley, CA.

Hastie, H., Merigaud, N., Liu, X and Oliver Lemon. (2010) " 'Let's Go Dude', Using The Spoken Dialogue Challenge to Teach Spoken Dialogue Development", SLT 2010, Berkeley, CA.

Raux, A., Langner, B., Bohus, D., Black, A., Eskenazi, M. (2005) "Let's go public! Taking a spoken dialog system to the real world", Interspeech 2005, Lisbon, Portugal.

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.

Thomson B., Yu, K. Keizer, S., Gasic, M., Jurcicek, F., Mairesse, F. and Young, S. "Bayesian Dialogue System for the Let's Go Spoken Dialogue Challenge", SLT 2010, Berkeley, CA.

Williams, J., Arizmendi, I., and Conkie, A. "Demonstration of AT&T 'Let's Go': A Production-Grade Statistical Spoken Dialog System." SLT 2010, Berkeley, CA.

Zhu, Y., Yang, Z., Meng, H., Li, B., Levow, G., and King, I. (2010) "Using Finite State Machines for Evaluating Spoken Dialog Systems", SLT 2010, Berkeley, CA.