DialPort, Gone Live: An Update After A Year of Development

Kyusong Lee, Tiancheng Zhao, Yulun Du, Edward Cai, Allen Lu, Eli Pincus[1], David Traum[1],
Stefan Ultes[2], Lina M. Rojas-Barahona[2], Milica Gasic[2], Steve Young[2] and Maxine Eskenazi

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA

[1]USC Institute for Creative Technologies, 12015 Waterfront Dr, Playa Vista, CA 90094, USA

[2]Department of Engineering, University of Cambridge, Cambridge, UK

{kyusongl,tianchez,yulund,edcai,arlu,max}@cs.cmu.edu

[1]{pincus, traum}@ict.usc.edu

[2]{su259,lmr46,mg436,sjy11}@eng.cam.ac.uk

Abstract

DialPort collects user data for connected spoken dialog systems. At present six systems are linked to a central portal that directs the user to the applicable system and suggests systems that the user may be interested in. User data has started to flow into the system.

1 Introduction

The goal of the DialPort spoken dialog portal is to gather large amounts of real user data for spoken dialog systems (SDS). Sophisticated statistical representations in state of the art SDS, require large amounts of data. While industry has this, they cannot share this treasure. Academia has difficulty getting even small amounts of similar data. With one central portal, connected to many different systems, the task of advertising and affording user access can be done in one centralized place that all systems can connect to. DialPort provides a steady stream of data, allowing system creators to focus on developing their systems. The portal decides what service the user wants and connects them to the appropriate system which carries on a dialog with the user, returning control to the portal at the end.

DialPort (Zhao et al., 2016) began with a central agent and the Let'sForecast weather information system. The Cambridge restaurant system (Gasic et al., 2015) and a general restaurant system (Let's Eat, that handles cities that Cambridge does not cover) joined the portal. A chatbot, Qubot, was developed to deal with out-of-domain requests. Later, more systems connected to the portal. A flow of users has begun interacting with the portal. Originally envisioned as a website with a list of the urls of systems a user could try, the portal has become easier to use, more closely resembling

what users might expect, given their exposure to the Amazon ECHO[1] and Google HOME[2], etc. In order to get a flow of users started, DialPort developers expanded the number of connected systems to make the portal offerings more attractive and relevant. They also made the interface easier to use. By the end of March 2017, in addition to the above systems, the portal also included Mr. Clue, a word game from USC (Pincus and Traum, 2016), a restaurant opinion bot (Let's Discuss, CMU), and a bus information system derived from Let's Go (Raux et al., 2005). The portal offers users the option of typing or talking and of seeing an agent or just hearing it. With few connected systems in previous versions it was difficult to assess the portal's switching mechanisms. The increased number of systems challenges the portal to make better decisions and have better a switching strategy. It also demands changes in the frequency of recommendations to connected systems. And it challenged the nature of the agent: some users prefer no visual agent; others couldn't use speech with the system.

A short history of DialPort DialPort started with a call for research groups to link their SDS to the portal and a website listing SDS urls for users to try out. It quickly evolved into one user-friendly portal where, all of the SDS are accessed through one central agent, users being seamlessly transferred from one system to another. System connections go through an API that sends them the ASR result (Chrome at present). The system was tried out informally (Lee et al., 2017) to determine whether the portal fulfilled criteria such as: timely response, correct transfer (to what the user wanted), and correct recommendation of systems (not saying for example, you can ask me about

---

[1]https://www.amazon.com

[2]https://madeby.google.com/home/

restaurants in Cambridge just after the user has finished talking to that system).

2 External Agents (ESes)

The first assessment of the interface (Lee et al., 2017) included five External Systems (ESes, that is, systems that are joined to the portal and are thus not part of the central portal - they can be from CMU as well as from other sites): Let'sForecast, Cambridge SDS on restaurants, Lets Eat; Mr Clue word game; and Qubot chatbot handling out of domain requests. Since then, Let's Go and Let'sDiscuss, a chatbot that gives restaurant reviews, have joined. The latter systems, by the CMU portal group, offer new services hoping to attract more diverse users and encourage them to become return users.

Cambridge The Cambridge restaurant information system helps users find a restaurant in Cambridge, UK based on the area, the price range or the food type. The current database has just over 100 restaurants and is implemented using the multi-domain statistical dialogue system toolkit PyDial (Ultes et al., 2017). To connect PyDial to Dialport, PyDial's dialogue server interface is used. It is implemented as an HTTP server expecting JSON messages from the Dialport client. The system runs a trained dialogue policy based on the GP-SARSA algorithm (Gašić et al., 2010).

Mr. Clue Mr. Clue plays a simple word-guessing game (Pincus and Traum, 2016). Mr. Clue is the clue-giver and the user plays the role of guesser. Mr. Clue mines his clues from pre-existing web and database resources such as dictionary.com and WordNet. Clue lists used are only clues that pass an automatic filter described in (Pincus and Traum, 2016). The original Mr. Clue was updated to enable successful communication with Dialport. First, since the original Mr. Clue listens for VH messages (a variant of ActiveMQ messaging used by the Virtual Human Toolkit (Hartholt et al., 2013), we built an HTTP server that converts HTTP messages (expected in JSON format) to VH messages. Second, since DialPort has multiple users in parallel, Mr. Clue was updated to launch a new agent instance for each new HTTP session (user) that is directed to the game from the main DialPort system. Mr. Clue is always in one of 2 states (in-game or out-game). The out-game state dialogue is limited to asking if the user wants to play another round (and offering to give instructions in the beginning of a session). The user can use goodbye keyword to exit the system at any time. This sends an exit message to DialPort and allowing it to take back control. For its 150 second rounds, timing information is kept on the back-end and sent to the front-end (DialPort) in every message. For each new session, the agent chooses 1 of 77 different pre-compiled clue lists (each with 10 unique target-words) at random. It keeps track of which lists have been used for a session so a user will never play the same round twice (for a given session).

Let'sDiscuss LetsDiscuss responds to queries about a specific restaurant by finding relevant segments of user reviews. It searches a database of restaurant reviews obtained from Zomato and Yelp. We formed a list of general discussion points for restaurants (service, atmosphere, etc). For each discussion point, a list of relevant keywords was compiled using WordNet, thesaurus, and by categorizing the most frequently words found in reviews.

Other Systems QuBot, a chatbot from Pohang University and CMU, is used for out-of-domain handling. Let'sForecast, from CMU, uses the NOAA website. Let's Eat from CMU is based on Yelp, finding restaurants for cities that Cambridge does not cover and for Cambridge if that system is down. Let's Go, derived from the Let's Go system (Raux et al., 2005), is based on an end-to-end recurrent neural network structure and a backend that covers cities other than Pittsburgh.

3 DialPort Platform

In informal trials, some aspects of the portal's interaction were not effective for some users. This included the use of speech (as opposed to typing), the use of a visual agent, the absence of both graphical and speech response, feedback and portal behavior. Some ES need graphics to supplement their verbal information. Since Mr Clue keeps score and timing of users' answers, its instructions and scores are shown on a blackboard. Let's Go shows a map with the bus trajectory from departure to arrival.

Feedback and communication The portal gives users feedback for: available topics, system state, and present system state. Skylar doesn't interrupt the dialog with a list of topics. Rather

171

it suggests one topic every few turns. This evenly steers users to all of the ES. A banner at the bottom of the screen reminds users of all the topics that can be discussed. Another box indicates the system state in order to avoid user confusion about who has the floor. It shows, for example, whether the system is processing the speech or is still waiting for them to talk. The box shows:

- idle (either from timeout or from the user clicking on the box to pause the system);
- listening (this is shown from the instant the ASR begins to process speech to when it is finished);
- speaking (from when the TTS begins output to when it is finished);
- thinking (from when the ASR output is sent to the NLU to when the DM issues its action).

Finally, the system informs the user of the present state of the dialog. Do you still want XX (e.g. Pittsburgh)? reveals that the user preference for Pittsburgh has not been used for a while, and Skylar's forgetting curve is ready to eliminate it. The dynamic choice of implicit or explicit confirmation covers the global dialog state.

3.1 Changes in the portal's behavior

As more ES join the portal, policies and strategies have become more flexible. There are two major changes to the portal's behavior: ES selection policy and ES recommendation policy. Starting with few ESes, each on very different topics, the agent selection policy simply tried to detect the topic in the users' request and select the corresponding ES. As more ESes connect to the portal, non-trivial relationships among ESes emerge:

1) *Dialog context sensitive agent selection:* The optimal choice of ES may depend on discourse history. For example, Let'sForecast, Cambridge restaurant and Let's Eat: after the user has weather information for city X, they say, recommend a place to have lunch. Choosing between Let's Eat and Cambridge restaurant depends on the value of city X, because Cambridge restaurant covers places to eat in Cambridge UK and Let's Eat covers other places.

2) *Discourse Obligation for Agent selection:* Users have various ways to make requests: request (tell me xxx), WH-question (what's the weather in xx) or Yes/No-question (Is it going to rain?). A natural dialog should answer a user according to the way in which they made their earlier requests (Traum and Allen, 1994). For example, the weather system should produce the natural Yes it's going to rain instead of a full weather report, for the third question above. We thus keep the user's initial request intent in the global dialog context and share it with the relevant ESes.

The recommendation policy has been improved in two ways: **1)** All participating system developers agreed that Skylar should give ES recommendations on a rotating basis so that all systems are recommended equally. Skylar no longer makes a recommendation at the end of each system turn. Recommendations are made about every four turns and, as mentioned above, are not for a system that the user recently interacted with. **2)** Fine grained recommendation: As more ESes joined the portal, we began to exploit the relatedness among ESs in order to generate more targeted recommendations. For instance, we tuned the policy to have a higher probability of recommending the Let'sDiscuss restaurant review function when users obtain restaurant information by prompting, do you want to hear a review about this place?

Finally, the NLU has been extended to support multi-intent multi-domain identification by reducing the problem to a multi-label classification task using a one-vs-all strategy. The weighted average F-1 score for multi-intent and multi-domain classification is 0.93.
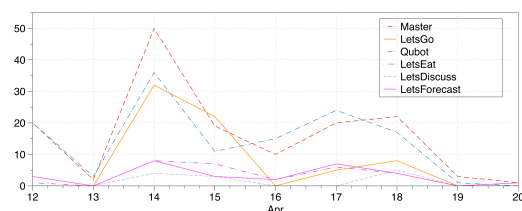

Figure 1: Number of dialog turns over time per ES

4 Go "Live"

There are several types of portal users. First, the developers themselves try out the system. Then they ask friends and family to try it. Users can be paid. Finally we have users who really need the information or gaming pleasure. We define two potential types of users (using IP addresses): explorers and real users. Explorers are trying the system for the first time. They explore several of the ESes, but they do not have any real gaming or information need. Real users have returned to use the por-

tal, asking for something they need or enjoy. They may speak to less of the ESes during their visit, but have some real. The first advertising attempt using Google AdWords[3] attracted few explorers and no real users. The following factors may explain why users did not have a dialog with the system: presence of human study consent form; not using Chrome browser (solved by making a typing-only version); user didn't want any portal services; user didn't have a microphone; user didn't understand the purpose of the portal (we gave Skylar an opening monologue explaining what the data is for).

4.1 Can DialPort collect data?

The AdWord experience lead us to published a Facebook page on April 12, 2017. The page was to attract both explorers and real users through both organic (friends and friends of friends) and paid distribution. Despite the short time (4-12 to 4-20) that it has been published, there have been a total of 51 dialogs (excluding all dialogs from participating research teams). As of April 20, DialPort spent about $52 in advertising to reach 1776 individuals getting 147 page views, 47 likes and 346 engagements (shares or clicks). About 40% of the clicks were from mobile devices as opposed to computers. This underlines the need for mobile versions of DialPort.

The average length of a dialog is 8.7turns (7.18 stdev) and 129.51s (stdev 138.03). There were 14.9% return users, although another person could be using that computer and some places have automatic IP assignment. 52.9% of the dialogs were spoken as opposed to typed. The average ASR delay was 925.03ms. On average, users tried 4.8 systems per dialog. The distribution of dialog turns per ES and for the portal over time is shown on Figure 1. Some systems are getting less use than others. This will be countered by paid advertising campaigns that promote each specific system.

5 Conclusion

This paper has presented a novel portal that collects spoken dialog data for connected systems. It has begun to collect data for the present seven systems. In order to improve service an audio server is under construction as are smartphone and tablet versions. The portal welcomes new external systems.

---

[3] https://adwords.google.com

References

M Gasic, Dongho Kim, Pirros Tsiakoulis, and Steve Young. 2015. Distributed dialogue policies for multi-domain statistical dialogue management. In *Acoustics, Speech and Signal Processing (ICASSP)*. pages 5371–5375.

Milica Gašić, Filip Jurčíček, Simon Keizer, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. Gaussian processes for fast policy optimisation of pomdp-based dialogue managers. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. pages 201–204.

Arno Hartholt, David Traum, Stacy C Marsella, Ari Shapiro, Giota Stratou, Anton Leuski, Louis-Philippe Morency, and Jonathan Gratch. 2013. All together now. In *International Workshop on Intelligent Virtual Agents*. Springer, pages 368–381.

Kyusong Lee, Tiancheng Zhao, Stefan Ultes, Lina Rojas-Barahona, Eli Pincus, David Traum, and Maxine Eskenazi. 2017. An assessment framework for dialport. *Proceedings of the International Workshop on Spoken Dialogue Systems Technology* .

Eli Pincus and David Traum. 2016. Towards Automatic Identification of Effective Clues for Team Word-Guessing Games. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*. European Language Resources Association, Portoro, Slovenia, pages 2741–2747.

Antoine Raux, Brian Langner, Dan Bohus, Alan W Black, and Maxine Eskenazi. 2005. Lets go public! taking a spoken dialog system to the real world. In *in Proc. of Interspeech 2005*. Citeseer.

David R Traum and James F Allen. 1994. Discourse obligations in dialogue processing. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 1–8.

Stefan Ultes, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Inigo Casanueva, Pawe Budzianowski, Tsung-Hsien Wen Nikola Mrksic, Milica Gasic, , and Steve J. Young. 2017. Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL Demo. Association of Computational Linguistics*.

Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. 2016. The dialport portal: Grouping diverse types of spoken dialog systems. *Workshop on Chatbots and Conversational Agents* .