Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

An Oral Exam for Measuring a Dialog System's Capabilities

David Cohen
Carnegie Mellon University
NASA Research Park, Bldg 23
Moffett Field, CA 94035
david.cohen@sv.cmu.edu

Ian Lane
Carnegie Mellon University
NASA Research Park, Bldg 23
Moffett Field, CA 94035
lane@cs.cmu.edu

Abstract

This paper suggests a model and methodology for measuring the breadth and flexibility of a dialog system's capabilities. The approach relies on having human evaluators administer a targeted oral exam to a system and provide their subjective views of that system's performance on each test problem. We present results from one instantiation of this test being performed on two publicly-accessible dialog systems and a human, and show that the suggested metrics do provide useful insights into the relative strengths and weaknesses of these systems. Results suggest that this approach can be performed with reasonable reliability and with reasonable amounts of effort. We hope that authors will augment their reporting with this approach to improve clarity and make more direct progress toward broadly-capable dialog systems.

Introduction

Two of the most glaring weaknesses of current dialog systems are their lack of breadth and flexibility. This already has negative practical consequences on system usability, and we can only expect the problem to get worse if these characteristics are not emphasized.

Breadth is a system's ability to perform adequately on a variety of reasoning tasks. A lack of breadth has negative consequences on user experience today. For example, if a system has advanced capabilities without foundational capabilities, it may not be able to fail gracefully. The following example dialog with Google Now[1] illustrates this:

User: How far is it to the moon?
Google Now: The moon is 238,000 miles from Earth.
User: How far is it from the moon to Saturn?
Google Now: about 1.4 billion km (*incorrect*)
User: How are you computing the distance between celestial bodies?
Google Now: *Shows Google search results: Celestial navigation - Wikipedia, ...*
User: Is the moon a thing?
Google Now: *Shows Google search results: 10 Things You Didn't Know About the Moon, 10 Strange Secrets of the Moon, Moon Facts, ...*

Flexibility is a system's ability to perform adequately on a variety of tasks which require nearly identical knowledge and reasoning. A lack of flexibility has negative consequences on user experience with dialog systems today as well. If a user does not know exactly how to phrase their request, or what the system's exact limits are, they may be unable to get the system to work. The following two examples from Google Now illustrate this:

User: How is traffic to San Francisco?
Google Now: There is currently heavy traffic from your location to San Francisco, it is 1 hour and 6 minutes by car.

User: How is traffic to San Francisco via I-280?
Google Now: *Shows Google search results: San Francisco - Yelp, Bay Area Traffic Report, ...*

As the number of highly-specialized functions integrated into dialog systems increases, other issues will likely present themselves. Capabilities will have to integrate with each other, or else strange behavior may result. For example, Google Now warns users to leave for their scheduled events in a way that satisfies travel-time constraints, but does not warn them if they schedule two events with travel-time constraints which are unsatisfiable. Users will also continue to demand increased flexibility as the number of different devices and input modalities grow.

There is an underlying problem with the engineering approach that leads to highly specialized capabilities, namely that there is no attempt being made at building general capabilities. Generality is the characteristic whereby performance on one intelligence-requiring task is predictive of performance at other intelligence-requiring tasks. This idea goes back to early work on psychometrics (Spearman 1928), which showed that human performance across many diverse tests of intelligence are correlated, and may be partially explained by a general intelligence factor, combined with more specific intelligences.

[1]All examples in this paper were from interactions during August or September of 2015. The dialog systems used are cloud-based, subject to A/B testing, and are updated over time. Therefore, individual examples may not be repeatable.

If engineers are to make progress toward generally-capable dialog systems, there will have to be a scientific process in place to support them. We believe that this process is currently too difficult for dialog systems researchers, and we make several contributions intended to kick-start it. 1) Researchers need a model for explicitly describing the 'landscape of capabilities' upon which dialog systems can be evaluated. We present such a model and one demonstrative instantiation of this model. 2) The complex capabilities, breadth, and flexibility of a full-fledged dialog system are difficult to describe, and in limited space, research authors have no good option for concisely reporting this information. We present a methodology and metrics for evaluating systems' performance within a given instantiation of our model, which will allow researchers to concisely report information about their systems that today goes unreported. 3) We carry out the proposed evaluation on several existing systems for demonstrative purposes, and present results along with analysis of the method's reliability and practicality.

Proposed Approach

We propose evaluation driven by an explicit model of a dialog agent's capability landscape. All the capabilities so encoded are externally observable behaviors, so this approach is agnostic about internal representations and processes. Even so, our model groups together capability categories which intuitively require related classes of knowledge or reasoning.

Capability Taxonomy

We model the evaluation landscape as a set of capabilities K and capability categories C where each individual capability $k \in K$ is a member of exactly one category $c \in C$. The capabilities and categories used in this paper are described in Table 1.

Capabilities are general classes of behavior, which need to be further specified to generate a single test problem. A test domain consists of a set Θ of parameters and a function $Param : K \times \Theta \rightarrow \{T, F\}$ which returns true if and only if the the capability accepts the domain parameter. In this paper, each $\theta \in \Theta$ (shown in Table 2) is either a physical object or an action verb with a simple one or two-word form.

Instantiating a Capability Taxonomy and Domain

The taxonomy model described above needs to be instantiated with capabilities and capability categories, and a domain scope for evaluation needs to be defined. Test reliability will depend on the clarity of capability definitions and documentation. Capabilities should be selected to reflect the variety of functions taxonomy designers desire a system to have. Capabilities are grouped into categories based on an assumption that they require very similar knowledge and reasoning capabilities. Our hope is that taxonomy specifications will be developed collaboratively and shared widely.

For this paper, we used a small number of domain concepts (shown in Table 2) several of which are central to advanced capabilities of interest to the authors. For example, Google Now can give directions to a nearby coffee shop, so coffee, drinking, cars, coffee shops, and driving are all concepts within the test domain. Test performance for an individual system will depend strongly on the domain, so it is important for researchers to report the domain they test in.

Elicitation Trials

The basic unit of testing is the elicitation trial, in which a human tester tries to get the system being tested to demonstrate the desired capability. The tester uses a tool which presents a capability for them to test along with the domain concept which parametrizes the capability. The tester then must formulate a problem in order to elicit the desired behavior. The tool presents documentation for the capability, which includes example eliciting utterances the tester might attempt. The tester has the option to skip a trial if they can not easily formulate a test, and the tester and system can both attempt clarification at the tester's discretion. The tester labels each trial with one of the following: 'Pass', 'Fail', 'Alternative'. Where Pass and Fail are self-explanatory, and Alternative indicates that the system performed some action that gives the user the right information, but fails to perform the capability desired, even after optional repeated attempts and clarifications.

An example elicitation trial may proceed as follows: The evaluation tool selects the 'Class Search' capability, and the 'coffee shop' physical noun. Documentation and example eliciting utterances for 'Class Search' are shown to the evaluator, along with the parameter 'coffee shop' and its definition. The human evaluator must then formulate their own test utterance and judge the response:

Human Evaluator: Do you know of any coffee shops?
Test System: *Shows search engine results: 13 Tips to Open a Successful Coffee Shop, Coffee Shop Tips, ...*

If the evaluator does not want to rephrase, the trial would end there, and the evaluator should score it as a 'Fail'.

Below is another elicitation trial where the evaluator decides that clarification is needed in order to score the system. In this case, the tool selected the 'Has-a YNQ' capability and the 'moon' physical noun:

Human Evaluator: Do moons have gravity?
Test System: Duh.
Human Evaluator: Yes or no.
Test System: Or none of the above?

The initial response was unclear, so the evaluator clarified the question. Once the system evaded the evaluator's second attempt, the human gave up and failed the system on that trial.

Both of the previous examples were one-off question-answering capabilities, and do not necessarily require a dialog system. The 'Narrowed Search' capability does require an extended dialog in order for the capability to be performed. For the following example, the tool selected the 'Narrowed Search' capability, and the 'moon' concept:

Capability Category	Capability	Example Elicitation	Example Correct Response
c_1 Physical Object Ontology	Hierarchy YNQ	"Is a car a type of vehicle?"	"Yes."
	Get Definition	"Define car."	"A car is a four-wheeled vehicle, typically powered by gasoline ..."
	Has-a YNQ	"Do cars have ears?"	"No."
c_2 Physical Object Search	Class Search	"What cars do you know of?"	"I have records for over 10K cars."
	Property Search	"What blue used cars do you know of?"	"There are 14 blue used cars in my database."
	Narrowed Search	..."Are any of them convertible?"	"None of the blue used cars are convertible."
	Alternative Search	..."How about red?"	"There are 12 red used cars in my database."
c_3 Action Verb Ontology	Effect YNQ	"If a person rides a bus, does he stay in the same place?"	"No."
	Pre-conditions YNQ	"Does a person need money to ride a bus?"	"Yes."
	During WHQ	"What happens while riding a bus?"	"The bus driver directs the bus along the bus route, stopping at scheduled stops."
c_4 Simple Planning	Provide Plan	"I want to go downtown"	"Take the 4b Bus."
	Grounded Effects YNQ	"If I ride the 4b, will it take me downtown?"	"Yes."
	Grounded Pre-conditions YNQ	"Can I get on the 4b bus if I am at 4th and Duran?"	"No."

Table 1: Summary of the example taxonomy of capabilities and capability categories used in the presented evaluations. Categories c_1 and c_2 are parametrized by physical nouns, c_3 and c_4 are parametrized by action verbs as shown in Table 2.

Physical Nouns	Action Verbs
moon, sports team, coffee shop, city, car	play game, drink, drive, pay

Table 2: The domain of evaluation used within this paper. These concepts are relevant to some advanced capabilities of interest to the authors, such as Google Now's navigation function.

Human Evaluator: What moons does Saturn have?
Test System: Saturn has moons Titan, Enceladus and others.
Human Evaluator: Do any of them have volcanic activity?
Test System: *Shows search engine results: Volcanology on Mars (Wikipedia), Volcano (Wikipedia), ...*

The system correctly performed the initial search, but was unable to narrow the search by adding the constraint that the moon should have volcanic activity, so it fails this trial.

Each trial t for system s is represented by the tuple: $t = (e[t], k[t], \theta[t], o[t])$, where $e[t]$ is the evaluator, $k[t]$ is the capability being tested, $\theta[t]$ is the parameter for the capability, and $o[t] \in \{0, 1\}$ corresponding to 'Fail' / 'Pass' is the outcome of the trial according to the evaluator ('Alternative' test results are mapped to 'Fail' for the purposes of all the metrics in this paper, though other metrics could be introduced to measure these). We also define $c[t]$ as the capability category the trial is testing (recall from earlier, $c \in C$ where C is the set of taxonomy concept categories), which can be determined uniquely from $k[t]$. After all evaluators have tested a system, the test result T_s is the set of trials used for analysis.

For each evaluator, the tool traverses the entire taxonomy, grouping together all tests from the same category. Domain concept parameters are selected randomly for each problem, meaning that there is a possibility of multiple problems having the same capability and parameter.

Evaluation Metrics

Evaluating Performance Performance P of a system s within a capability k is computed by the following:

$$P_k(s) = \text{Average}(\{o[t]; t \in T_s, k[t] = k\})$$

Performance P of a system s within a capability category c is the average capability performance of capabilities within that category:

$$P_c(s) = \text{Average}(\{P_k(s); k \in c\})$$

Total performance is the sum of category performance measures over all categories:

$$P_{\text{total}}(s) = \sum_{c \in C} P_c(s)$$

Evaluating Breadth Breadth should measure the ability of the system to make use of a wide variety of reasoning procedures and knowledge. In terms of our capability taxonomy model, this means breadth should measure the diversity of capability categories which can be performed. Measuring diversity is in itself a complicated problem, leading to several approaches such as that introduced by (Rao 1982) and investigated in (Pavoine, Ollier, and Pontier 2005). For simplicity, we measure diversity using the normalized Shannon entropy measure (Kumar, Kumar, and Kapur 1986), denoted by $\widetilde{H}(Q)$ where Q is a discrete probability distribution.:

$$\widetilde{H}(Q) = -\frac{1}{\ln(n)} \sum_{i=1}^{n} q_i * \ln(q_i)$$

A system's breadth B is the diversity of capability category performance:

$$B(s) = \begin{cases} 0 & \text{if } P_{\text{total}}(s) = 0 \\ \widetilde{H}(Q_B) & \text{otherwise} \end{cases}$$

where the distribution Q_B has the probability mass function:

$$q_B(c) = \frac{P_c(s)}{P_{\text{total}}(s)}, c \in C$$

The MLE estimate of entropy is negatively biased as described in (Paninski 2003), but estimates have low variance, so $\widetilde{H}$ serves the purpose of a reliable diversity measure. However, comparisons should only be made between breadth measures based on the same number of samples, due to the estimator bias's dependence on sample size.

Evaluating Flexibility Flexibility should intuitively measure the ability of the system to bring some knowledge and reasoning to bear in a variety of ways. As such, we compute flexibility of a capability category F_c as the diversity of capability performance within that category.

$$F_c(s) = \begin{cases} 0 & \text{if } P_c(s) = 0 \\ \widetilde{H}(Q_F(c)) & \text{otherwise} \end{cases}$$

where the distribution $Q_F(c)$ has the probability mass function:

$$q_F(k) = \frac{P_k(s)}{\sum_{k' \in c} P_{k'}(s)}, k \in c$$

This measure will distinguish between systems with one highly-specialized capability and systems with several.

Procedure

We perform evaluations on two computer dialog systems and one human within a small but diverse test domain (shown in Table 2). The two computer dialog systems were Google Now, which uses a speech input interface, and Cleverbot, which uses a typed interface. The human evaluation was performed anonymously over Internet chat, where the test subject did not have access to Internet search. The human test subject was a graduate student in our lab, and is not meant to be representative of the average human. We use six evaluators, who evaluate the test systems in counter-balanced order.

Each evaluator was given a basic explanation of the purpose of the experiment, and shown instructions for how to rate responses and an example. They were then shown the tool, and supervised through the first couple problems. One of the experimenters was available at all times to answer their questions. Each evaluator performed two elicitation trials per capability per test system. Evaluators performed the test on one system completely before moving on to the next system, and evaluated both programs and the human in counterbalanced order in one 1.5 hour session. In total, this process created 468 scored problems for our analysis.

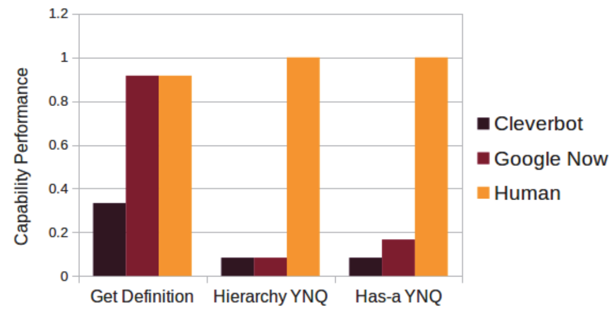

Figure 1: Google Now shows a lack of flexibility by performing unevenly across capabilities that have similar knowledge requirements. This difference is captured by the flexibility metrics in Table 4. This table shows system capability performances (see definition of P_k) for the three capabilities k in category c_1, "Physical Object Ontology".

Experimental Results

This paper tries to evaluate three claims about our approach. First, we claim that our model and metrics can distinguish between systems with a small number of highly-specialized capabilities and systems with general capabilities. Second, we claim that this evaluation method gives reliable results across different test administrations. Lastly, we claim that this evaluation can be performed within a time frame typical of dialog system evaluations.

Exemplifying evidence for our first claim, Figure 1 shows a typical outcome illustrating the problem with lack of flexibility in Google Now. For "Get Definition", Google Now performs as well as a human, but this is a specialized implementation, and the knowledge which should contribute to "Get Definition" can not be used flexibly to answer questions which are conceptually very similar. Not only does the human outperform the competition on average (which is captured by performance numbers P_{c_1}, shown in bold in Table 3), human performance is more consistent across capabilities within the category (which is captured by the flexibility numbers F_{c_1}, shown in bold in Table 4).

To provide evidence for our second claim, we start by reporting standard deviations for our performance metrics in Table 3. These tests are only consistent enough to support drawing slightly significant conclusions about the differences between Google Now and Cleverbot for the Physical Object Ontology and Simple Planning categories (c_1 and

c_4, $p < .1$ with Holm-adjusted t-test). A slightly different approach is used to measure and determine reliability for the entropy-based metrics of breadth and flexibility. Since a sample size much larger than the number of bins is required for favorable properties of entropy estimators (such as normality) to materialize (Paninski 2003), we base our measures on data from groups of three judges. We estimate our test-retest standard deviation by averaging the sample standard deviations taken from all possible pairs of non-overlapping groups of three judges. Results are shown in Table 4. Even with groups of three judges, variances in breadth and flexibility are too high to draw significant conclusions about differences between the two computer systems.

Measure	Cleverbot	Google Now	Human
P_{total}	0.31 (0.16)	1.04 (0.25)	3.63 (0.27)
P_{c_1}	**0.17 (0.1)**	**0.39 (0.16)**	**0.97 (0.06)**
P_{c_2}	0.06 (0.1)	0.21 (0.12)	0.79 (0.19)
P_{c_3}	0.06 (0.08)	0.11 (0.12)	0.92 (0.13)
P_{c_4}	0.03 (0.06)	0.33 (0.22)	0.94 (0.08)

Table 3: The human subject performance dominated Google Now and Cleverbot across all categories. A breakdown for c_1, "Physical Object Ontology", is shown in Figure 1.

Measure	Cleverbot	Google Now	Human
B	0.71 (0.2)	0.91 (0.05)	1 (0)
F_{c_1}	**0.54 (0.21)**	**0.48 (0.25)**	**1 (0)**
F_{c_2}	0 (0)	0.59 (0.09)	0.99 (0.01)
F_{c_3}	0.13 (0.13)	0.44 (0.44)	1 (0)
F_{c_4}	0 (0)	0.65 (0.18)	1 (0)

Table 4: The human subject demonstrated greater overall breadth and flexibility across all capability categories. A breakdown for the capabilities in c_1, "Physical Object Ontology", is shown in Figure 1.

To assess our third claim, we timed evaluators while instructing them to take their time and be accurate. Each elicitation trial took on average 62 seconds, meaning a single evaluator can comfortably test a system in 30 minutes using the taxonomy version and exam size used here. This should allow evaluation well within the time frame typically required for current dialog system evaluations with human subjects such as (Jordan, Albacete, and Katz 2015; Pincus, Georgila, and Traum 2015).

Prior Work

Dialog system evaluations are as old as AI itself. This section presents an overview of prior approaches, which, for the various reasons described, have not enabled clear reporting of system capabilities and have not led to consistent progress toward broadly capable and flexible systems.

Evaluating Chatbots

The Turing Test (Turing 1950) is the most famous test for computer intelligence, and it assumes a chatbot interface. If a computer can hold a conversation with a human (from another room, using a keyboard to communicate), whether it 'thinks' or not is irrelevant. The view that intelligence should be evaluated by observing behavior, rather than insisting on some unobservable internal phenomenon such as 'thought' is widely held today. (Cohen 2005) provides several useful criticisms of and alternatives to the Turing Test. First, the author argues that it is not diagnostic or specific enough to be used as a benchmark for incremental progress. Also, the author argues that a system that could pass the Turing test would not necessarily generalize to perform well on other intelligence-related tests. Nevertheless, the Turing Test is iconic, and pursued in the annual Loebner competition [2].

(Shawar and Atwell 2007) describes a small-scale evaluation of several chat-bots and attempts to provide more useful information than the Loebner competition. They test a machine-learning approach to learning response templates based on input patterns by generating 3 versions of the ALICE chatbot system[3], and analysing the reasonableness of a chat system's responses, relatedness of an information-retrieval system's results, and the ability of the search system to find answers. This is important progress relative to the sole Loebner / Turing criterion of 'acting human', but definitions need to be more precise, test reliability needs to be better understood, and performance needs to be broken down by capability. (Morrissey and Kirakowski 2013) clarifies the Loebner criterion by using human evaluator surveys and PCA to identify four factors contributing to chatbot naturalness; they dub these factors conscientiousness, manners, thoroughness, and originality. These factors could be adapted as categories of capabilities in the approach we take here, where they would form part of a larger comprehensive landscape of capabilities. (Vinyals and Le 2015) presents a novel chatbot architecture, and uses human evaluators to compare systems by how crowd-sourced human evaluators prefer their responses to questions. However, there are no clear criteria for selecting questions, and no analysis is presented of what these questions are testing or how systems differ in their performance depending on the types of questions asked.

Evaluating Spoken Dialog Systems

Most spoken dialog system evaluation has focused on user satisfaction and task completion. The most influential such approach is the PARADISE framework (Walker et al. 1997). PARADISE presumes that maximizing user satisfaction is the purpose of dialog systems, and goes about defining an approach for predicting user satisfaction based on more readily-available performance and cost metrics. There are several problems with this assumption and any particular attempt to apply PARADISE. Firstly, dialog systems may serve purposes other than satisfying their users for example, a tutoring system may be designed to improve test scores in its users whether they like it or not, a testing system may be designed to evaluate a user's knowledge, and a debate system may be designed to frustrate and embarrass its

[2]http://www.loebner.net/Prizef/loebner-prize.html
[3]http://alice.pandorabots.com

users. Secondly, even if human satisfaction was the end goal, we do not believe that dialog systems have reached a level of breadth where focusing on performance now will yield long-term benefits. Lastly and more practically, task success and other variables which will contribute to user satisfaction are highly dependent on experimental conditions, including what exact tasks users attempt to use the system for and what sort of highly-suggestive help the system gives the user.

While PARADISE relies on the user's own judgement of the system's performance, there has also been substantial work on exclusively using annotations by third parties to evaluate dialog systems. Both expert annotators as in (Dayanidhi et al. 2013), and novices via crowd-sourcing as in (Yang et al. 2010) have been successfully applied. The only evaluation we are aware of that attempted to include expert judgement by actually having experts interact with the test system was in (Scheffler, Roller, and Reithinger 2009), but this was only to test two capabilities of the spoken interaction. Though crowd-sourcing has been used effectively for many NLP annotation tasks (Snow et al. 2008), it is unproven for creative and technically demanding tasks such as the evaluation we propose.

Question-Answering to Evaluate Language Understanding

A common early approach to evaluating story understanding systems was to ask the systems questions about the story after reading, such as in (Dyer 1982). This approach also reflects the intuition that 'understanding' as an internal process is ill-defined and behavior is a better way to judge an intelligent system. A similar approach was used to demonstrate understanding and intelligence in early dialog systems such as SHRDLU (Winograd 1971). However, questions and passages were often selected to demonstrate rather than to realistically evaluate capabilities. Also, by having the system designer select questions, issues of performance with non-expert users and system flexibility were ignored, and the practical capabilities of systems were greatly exaggerated.

The Text Retrieval Challenge (TREC) conference (Voorhees and Buckland 2003) has yielded consistent improvements to document and answer extraction from large text corpora. Though a large amount of factual knowledge can be demonstrated through these techniques, the TREC challenge does little to test a system's ability to synthesize knowledge (a.k.a reasoning). Another question-answering proposal is described in (Levesque 2012). This involves a specific type of question called a Winograd Schema, which is designed to require application of knowledge and defeat shallow methods. Interestingly, the author makes a claim similar to Turing's, that a system which can pass a Winograd Schema test is thinking. The author mentions the importance of having a test that permits incremental progress, but his only suggestion is that questions can have varying levels of background knowledge requirements, without being specific about what those different types of knowledge might be.

(Levesque 2012) points out that a problem with dialog-based evaluations is the trickery and evasion which dialog systems can use to avoid being tested on their knowledge. Our approach specifically is meant to counteract this weakness by targeting specific capabilities, and failing the test-taker if they do not perform them. At the same time, using dialog allows this approach to avoid several inherent weaknesses of question-answering evaluations; for example, many possible answers could be correct but unanticipated by test set designers, answers might require follow-ups for clarification, or questions might require disambiguation.

Psychometric Intelligence Tests

Psychometric research has yielded interesting models and insights relevant to test design for measuring human intelligence. Pearson's seminal early work (Spearman 1928) laid much of the conceptual and statistical foundation for intelligence testing. This has been further refined and adapted to specific aptitude tests with item response theory (IRT) (Embretson and Reise 2013). The entire concept of test reliability depends on the assumption that there is some underlying factor being measured. For general intelligence tests, this is widely known as the g factor (Spearman 1928), but in the case of highly specialized AI systems being tested on broad capabilities, it is difficult to argue that there is any such thing as an underlying general intelligence. Psychometric AI as discussed in (Bringsjord 2011) is the field dedicated to building computer programs which can perform reasonably well "on all established, validated tests of intelligence and mental ability", or in other words, computer programs which have general intelligence. One example test of intelligence is the Wechsler Adult Intelligent Scale (WAIS) (Pearson Education Inc. 2008), which is subdivided into performance subtests, much as our approach proposes. Some of the verbal subtests of WAIS bear a strong resemblance to the example capabilities we introduce in this paper.

Conclusion and Future Work

This paper presents a model and metrics which clarify the notions of breadth and flexibility, and a methodology for measuring dialog system capabilities and reporting them concisely. We urge dialog system researchers to incorporate this method, since doing so will greatly help readers understand what the presented systems can do without requiring extensive description.

The taxonomy used in this paper is demonstrative, and is focused on areas of capabilities which are of interest to the authors. To better represent broad research interests will require a standardization effort and collaboration among dialog system researchers.

Beyond using our approach to round out dialog system papers, an approach like ours may prove useful for human-level intelligent systems research. One problem with past psychometric AI evaluations is that IQ tests can be automated quite easily without building a generally intelligent system, as in (Sanghi and Dowe 2003; Evans 1964). Breadth and flexibility metrics should prove useful for detecting impostor programs such as these, as well as other programs which specifically target narrow capabilities.

References

Bringsjord, S. 2011. Psychometric artificial intelligence. *Journal of Experimental & Theoretical Artificial Intelligence* 23(3):271–277.

Cohen, P. R. 2005. If not turing's test, then what? *AI magazine* 26(4):61.

Dayanidhi, K.; Evanini, K.; Hunter, P.; Liscombe, J.; Pieraccini, R.; Suendermann, D.; and Gorelov, Z. 2013. System and method for robust evaluation of the user experience in automated spoken dialog systems. US Patent 8,520,808.

Dyer, M. G. 1982. In-depth understanding. a computer model of integrated processing for narrative comprehension. Technical report, DTIC Document.

Embretson, S. E., and Reise, S. P. 2013. *Item response theory*. Psychology Press.

Evans, T. G. 1964. A program for the solution of a class of geometric-analogy intelligence-test questions. Technical report, DTIC Document.

Jordan, P.; Albacete, P.; and Katz, S. 2015. Exploring the effects of redundancy within a tutorial dialogue system: Restating students responses. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 51.

Kumar, U.; Kumar, V.; and Kapur, J. N. 1986. Normalized measures of entropy. *International Journal Of General System* 12(1):55–69.

Levesque, H. J. 2012. The winograd schema challenge. In *KR*.

Morrissey, K., and Kirakowski, J. 2013. realness in chatbots: Establishing quantifiable criteria. In *Human-Computer Interaction. Interaction Modalities and Techniques*. Springer. 87–96.

Paninski, L. 2003. Estimation of entropy and mutual information. *Neural computation* 15(6):1191–1253.

Pavoine, S.; Ollier, S.; and Pontier, D. 2005. Measuring diversity from dissimilarities with rao's quadratic entropy: Are any dissimilarities suitable? *Theoretical Population Biology* 67(4):231 – 239.

Pearson Education Inc. 2008. Wechsler adult intelligence scale – fourth edition.

Pincus, E.; Georgila, K.; and Traum, D. 2015. Which synthetic voice should i choose for an evocative task? In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 105.

Rao, C. 1982. Diversity and dissimilarity coefficients: A unified approach. *Theoretical Population Biology* 21(1):24 – 43.

Sanghi, P., and Dowe, D. L. 2003. A computer program capable of passing iq tests. In *Proc. 4th ICCS International Conference on Cognitive Science (ICCS03)*, 570–575.

Scheffler, T.; Roller, R.; and Reithinger, N. 2009. Semi-automatic creation of resources for spoken dialog systems. In Mertsching, B.; Hund, M.; and Aziz, Z., eds., *KI 2009: Advances in Artificial Intelligence*, volume 5803 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 209–216.

Shawar, B. A., and Atwell, E. 2007. Different measurements metrics to evaluate a chatbot system. In *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, 89–96. Association for Computational Linguistics.

Snow, R.; O'Connor, B.; Jurafsky, D.; and Ng, A. Y. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, 254–263. Association for Computational Linguistics.

Spearman, C. 1928. The abilities of man; their nature and measurement.

Turing, A. M. 1950. Computing machinery and intelligence. *Mind* 433–460.

Vinyals, O., and Le, Q. V. 2015. A neural conversational model. *CoRR* abs/1506.05869.

Voorhees, E. M., and Buckland, L. 2003. Overview of the trec 2003 question answering track. In *TREC*, 54–68.

Walker, M. A.; Litman, D. J.; Kamm, C. A.; and Abella, A. 1997. Paradise: A framework for evaluating spoken dialogue agents. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, 271–280. Association for Computational Linguistics.

Winograd, T. 1971. Procedures as a representation for data in a computer program for understanding natural language. Technical report, DTIC Document.

Yang, Z.; Li, B.; Zhu, Y.; King, I.; Levow, G.; and Meng, H. 2010. Collection of user judgments on spoken dialog system with crowdsourcing. In *Spoken Language Technology Workshop (SLT), 2010 IEEE*, 277–282. IEEE.