ANTIK//Antikythera: Investigating Negotiated Language Making

Erik Stayton
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
estayton@mit.edu

Abstract
The ANTIK//*Antikythera* projects are an intervention in the experimental anthropology of human-machine interactions. This paper describes the theoretical grounding of the ANTIK adaptable natural-language interface in linguistics and artificial intelligence. It documents the state of the ANTIK system, which uses a machine-translation model to associate natural-language words and phrases with computer-language atoms and expressions. The paper also describes the design process of the *Antikythera* game, built around ANTIK, through prototyping, development, and user testing. It relates the key lessons learned about users' interaction styles, identifying and describing the "search for syntax" that characterizes interactions. Finally, the paper identifies future work on both parts of the ANTIK//*Antikythera* system.

Author Keywords
natural-language interaction; artificial intelligence; stimulus-response; behavior; learning; interfaces

Language and AI

Meaning-based theories favor a concept of teaching machines to communicate by developing the “transformational grammar” that translates between some surface structure of words into a deep structure of meaning. Communication becomes the transmission of meanings via correctly-structured strings of words.

Behavior-based teaching, in its simplest form, reflects Skinner's stimulus-response learning, and we apply this teaching method in the ANTIK system [9]. The user trains the machine learning system to perform certain actions by reinforcing them.

Machine Learning may involve neural, connectionist approaches inspired by Hebb [4]. Or, as here, it may involve purely statistical approaches without pretense of biological analogies. In either case, ML offers the promise of reducing the human labor of building rule sets, and allowing systems to adapt to live inputs.

Introduction

ANTIK//*Antikythera* is a research tool for the experimental anthropology of human-machine interactions. We investigate the interactional and perceptual aspects of the negotiation of machine behavior. *Antikythera* is a game about an ancient Greek artificial intelligence (AI). The player must solve a series of puzzles, or “diagnostics,” by teaching the AI system about its world. *Antikythera* is built on top of the ANTIK system, an adaptable natural-language interface (NLI).

Theoretical Background

Theories of Language

Chomskyan linguistics has been influential in computation as early as Chomsky's *Syntactic Structures* [3]. This influence continues despite the controversy around his more recent Minimalist Program. Both involve correspondences between sound or syntactic structures (patterns of words) and semantics (meanings).

Other theories of language complicate this notion. Anthropology and cultural studies have recognized strong ritual and phatic components of language [6] [2]. Much of what we say on a daily basis (e.g. “what's up”) is not intended to transmit *meanings*, but to produce socially-sanctioned *responses*.

Rather than treating language as a fixed system of statements, each having a particular meaning that can be determined, we treat language as a system for producing mutually agreed-upon *behaviors*. The user negotiates a language system by training ANTIK's adaptable NLI to perform tasks. Language is oriented toward producing state changes in the environment.

History of AI

ANTIK responds to both Good Old-Fashioned AI (GOFAI) and machine learning (ML) approaches. GOFAI, represented by the work of Minsky, Newell, Simon, and others, attempts to create universal logical structures to govern behavior [7] [8]. While attractive in theory, it has not been found possible to build comprehensive logical structures outside of confined environments. In contrast to GOFAI, ML approaches apply trained statistical classification (see sidebar).

ANTIK combines a small number of human authored rules with an ML system that can extend those rules. Unlike consumer NLI systems like Siri, ANTIK makes the training process explicit. We take an interactionist approach to AI design, seeking to create a plastic system that can react to its environments, and which makes use of human knowledge to operate more effectively [5].We focus on action in conditions of uncertainty, not theoretical perfection.

Technical Details

ANTIK is built in Pharo, an open-source implementation of Smalltalk. Pharo is an object-oriented language with graphical environment. ANTIK consists of a graphical user interface (GUI) with a text input, display window, and several buttons (Fig 2).

ANTIK processes natural-language text into Pharo commands. The prototype used a K-nearest neighbors algorithm (KNN) to associate user text strings with stored stimulus-response (S-R) pairs. But KNN cannot generate commands not in its training set. Machine-translation (MT) models can. ANTIK now applies a classic IBM Model 1 MT model [1]. This model provides good base-line behavior for simple tasks, and has a

Design and Testing Process

Early Development: Building the ANTIK system, generating paper prototypes for puzzles.

Prototype Testing: Paper prototypes tested with 4 users.

Development: Puzzles and pacing improved. Puzzles built into Pharo system.

Testing I: Author QA. Pharo system tested with 3 users.

Development: User manual created, memories developed, and teleprinter features added. New puzzles added.

Testing II: Author QA. Pharo system tested with 1 user.

Development: Manual improved, puzzles streamlined.

Testing III: Pharo system tested with 3 users.

large body of literature documenting improvements.

We use specially authored parallel corpora of English text and Pharo expressions to associate stimuli (user commands, as text) with responses (objects, methods, and arguments). The system makes associations using the expectation-maximization (EM) algorithm. The EM algorithm generates tables of likelihoods for the association of each English word with each Pharo atom. Stimuli (strings) are then translated back into responses (expressions) using these likelihoods, subject to the structural constraints of valid Pharo expressions. Valid expressions are produced by a synthesizer with access to all objects, methods, and arguments known to the system. The most likely response to any stimulus is chosen by default. The user reinforces correct responses, which adds an S-R pair to the training corpus. Incorrect responses are corrected by selecting and reinforcing the correct S-R pair.

Designing *Antikythera*

The *Antikythera* game world consists of a series of interfaces (computationally modeled lists, switches, dials) that are connected to the ANTIK system. Each such interface represents a diagnostic of a fictional Antikythera device. The player must complete the diagnostics to access the device's memory, and learn about its past. Recovered memory snippets provide a plot, and give the player a reason to teach the system.

A paper manual provides general information, a walkthrough of one whole puzzle and several other examples, a code reference, and a list of diagrams that provide texture and hints to the diagnostics. Players interact using textual commands. A player might type "add fire" into the ANTIK system, while attempting to solve the ELEMENTS diagnostic. This might be associated via the MT model with the object #elements, the method #add, and the arguments list #('fire'). Associations are built from trained data, so the user could equally teach the system that "foo bar" is associated with that Pharo expression. Or, the user could re-train the system so that "add fire" is instead associated with the object #colors, the method #add, and the arguments #('white').

Testing and Feedback

We performed multiple rounds of user testing to develop the system (see sidebar). We observed users, asked them to think aloud, and asked semi-structured follow-up questions. The *Antikythera* game has developed from a paper prototype of a single puzzle to a set of 7 computerized diagnostics. Early puzzles involved adding, removing, or selecting items from lists. Later puzzles involve new mechanics, such as blending colors, rotating dials, and applying appropriate senses to objects. These puzzles developed with user feedback,

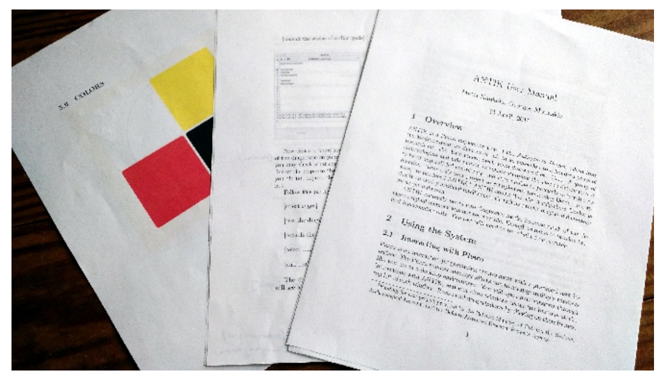

Figure 1: The paper manual for ANTIK, showing part of the walkthrough, and the appendix entry for the COLORS diagnostic.

Test-Driven Changes

Early user testing determined that foregrounding the computational nature of the game was compelling to players. This feedback drove us to include the narrative memory elements. It also pushed us to develop a teleprinter mechanic for displaying the system's supposed memories.

Users' engagement with cognitive aids, including pencil and paper and smartphones, seemed succesful in early tests. Inspired by this, we developed further puzzles requiring the use of such aids. DIAL requires the use of a protractor to identify appropriate rotation angles. COLORS favors online research into Greek color theory. These external interactions are ancillary to the core questions of this project. But they seem effective in allowing more obscure puzzles, thereby providing a sense of mystery that motivates interactions.

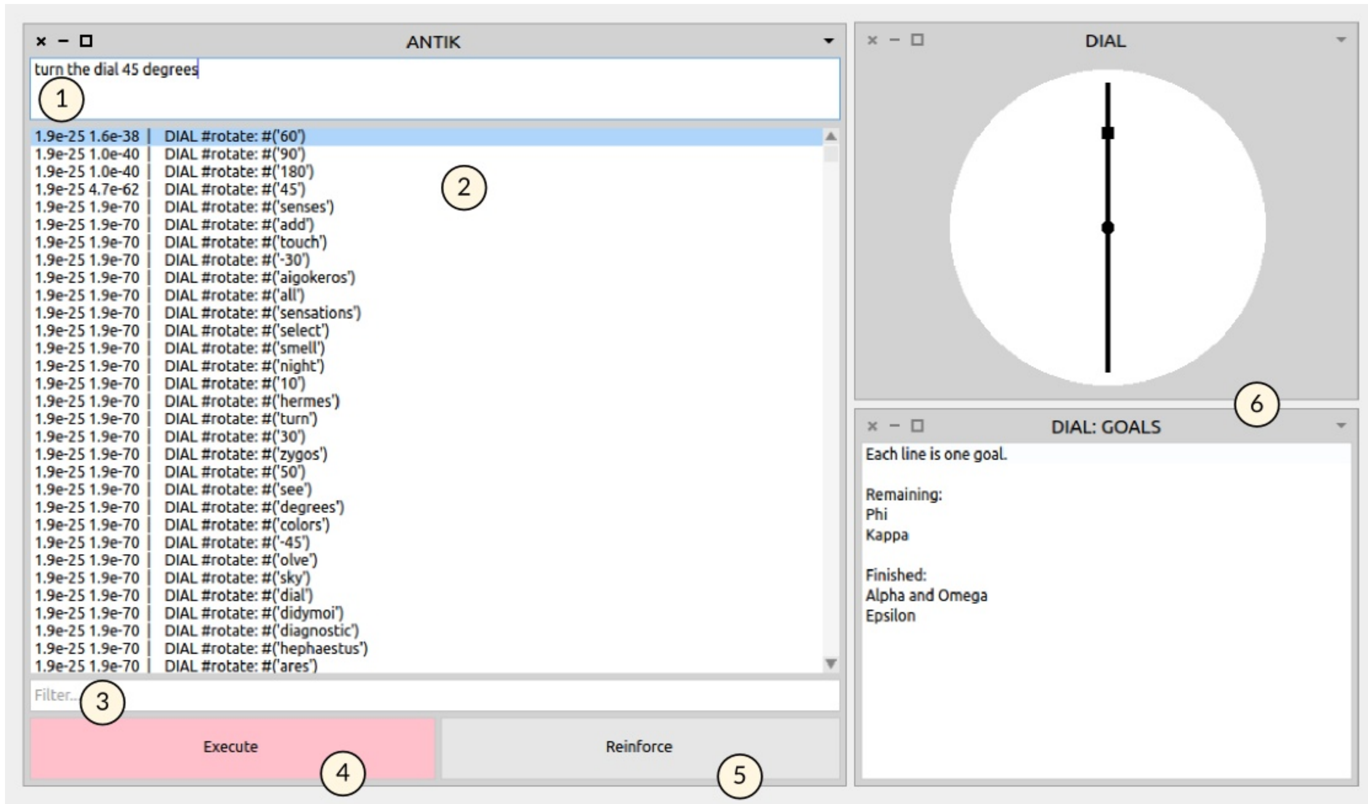

Figure 2: The ANTIK interface contains (1) command entry, (2) a list of probable responses, (3) aresponse filter, (4) a button to execute the selected response, (5) a button to train the stimulus-response association, and (6) space for other windows.

both to increase the variety of language the player had to use as well as to provide creative challenges.

User testing demonstrated that there was no clear model for how to interact with the system. We developed the manual and step-by-step walkthrough of the first diagnostic to address the resulting confusion.

Research Findings

Players were generally successful interacting with the AI system. Most of the difficulties encountered were part of adapting to a strange computing environment.

The commands users taught provide insights into their language-building. Of the 72 commands trained by

Commands

Of the 14 commands with errors, 2 referenced the wrong object, and 6 taught reasonable commands that are however disallowed by design.

Of the remaining 6: 1 represented a reasonable English request mapped to a bad Pharo command; 1 represented a reasonable request, not allowed by the system, mapped to a bad Pharo command; 3 represented nonsensical English requests mapped to reasonable Pharo commands; and
1 represented a seemingly nonsensical English phrase mapped to a meaningless Pharo command.

Of the commands that were somehow in error, several were due to mistakes in the use of the interface (not checking the response field carefully enough before clicking Reinforce). Two represent misapplications of concepts from one puzzle to another. Only 3 represent true failures of understanding.

users, 58 had matching English stimuli and Pharo commands, and were reasonable in the context of the system. Of these, 11 use the commands for purposes not anticipated by the designer (see sidebar).

The command logs show players transferring understanding from one puzzle to another. But those commands are sometimes inappropriate to the system. Playtester 2 followed the early examples exactly, and attempted to use the word “select” everywhere, including to refer to addition of items, and to operations that were clearly not selection (e.g. rotation). This is despite knowing that “add” existed in the system.

Playtesters 2 and 3, in following the examples very closely, generally left off the names of diagnostics from commands (“select athena” rather than “select athena gods”). By contrast, Playtester 1 tended to specify the name of the diagnostic: 8 of their 23 trained commands contain the diagnostic name unnecessarily. When asked why, s/he replied that it was an attempt to reduce ambiguity in case the same base command was used with a different diagnostic. (The author has also used this phrasing as a memory aid while testing.)

This difference appears to stem from a difference in experience: Playtester 2 is less experienced with text-based interfaces, and stuck to the examples thinking they were the only correct way to use the system. Playtester 3, also inexperienced, kept to the examples, and only at the end discovered that order of words did not matter. Playtester 1, more experienced with computer systems, expected to find syntax in the form of #method #argument #object. These varieties in language are important research findings for systems of this type. Unlike the author and an external expert, none of the playtesters shortened words. The only exception was one typo of “flow” for “flower.”

The first response of all playtesters to errors was to change the phrasing of their request. Often, players would pause, re-examine the command, look at the manual, re-type the command, and re-test. After entering equivalent commands and seeing the same inaccurate result, they would then remember to train the system. This seems natural: experience with computers has conditioned us to expect the need to use particular formats. In a strange environment like this one, when the system errs, we first doubt ourselves.

Players did improve in their willingness to train the system and their ability to handle errors. Playtester 2 initially reinforced answers only when the system was incorrect. But after several hours, s/he sometimes used reinforce even when the system had already picked the right answer. S/he explained that s/he wanted to avoid having to look through the responses list repeatedly.

Together, players' tendency to doubt themselves first, and to phrase requests in ways colored by their previous computing experience, suggests a general interaction principle we call the “search for syntax.” NLIs that are directly adaptable to the user are rare. Players interacting with the ANTIK system, operating in a strange space, tried to identify a syntax the system would respond to, whether that was by totemic use of “select”, or by drawing from previous experience with bash and SQL. In the first situation, the playtester rearranged symbols without worrying about meaning, based on the simple fact that the command appeared to make the right thing happen. In the second case, an understanding of the generally fixed format of

Perception of Competence

Despite some confusion, and needing to train the system to do new things, players' perception of ANTIK was generally one of competence.

All playtesters were positive in their responses to the statement “ANTIK understands what I want.” It got close enough, often enough, that it appeared competent.

Players were not as enthusiastic about the statements “I can train ANTIK successfully.” or “I see value to training ANTIK.” The importance of training the system only became clear after several hours of play, and even then was somewhat uncertain.

But all playtesters were very enthusiastic about their satisfaction with (eventually) reaching an understanding of the system. The setup of *Antikythera* was successful in making players feel like they had accomplished something worthwhile, even if they were only teaching ANTIK to add words to lists.

computational commands led to a search for the magic combination of object, method, and argument that would cue the desired effect. For both, meaning was subordinate to the demands of symbolic processing.

Our attempt at fostering negotiated communication, based on goals and behaviors rather than meaning, highlights a parallelism of human and machine: both respond to trial-and-error stimuli, and use them to structure communication. This search for syntax is an important consideration for the continued development of adaptable NLIs. More work on how to provide users with the tools to structure their own syntaxes rather than searching for them in the system, is vital.

Future Work

Research findings so far are limited to participant observation and brief semi-structured interviews with play-testers. No experiments have yet been conducted.

There are several directions for future work. First, A/B testing would improve the specificity of research findings. An A/B test may be conducted in which the A group uses the current ANTIK system, and the B group uses a system that disallows training, but contains all necessary commands pre-trained. This could identify how training specifically, as opposed to the rest of the system, effects perceived competence. The same A/B test could also compare users' emotional responses.

Second, interface changes might address the user's “search for syntax.” The Reinforce button might be made more prominent in the interface and the manual. Altering the AI to use a more conversational style, asking “Do you mean . . .?” instead of showing a list of possible responses, is another approach to explore.

Third, the ANTIK system will be integrated with other Pharo projects, including a digital humanities project for which an adaptable NLI would be useful. The integration task should provide new insights in this new context.

Acknowledgements

We thank all volunteers who provided user testing feedback, and classmates in CMS.846 who provided responses to presentations and drafts. We also thank Evan Donahue for his continued intellectual support, development assistance, and his work on related Pharo projects with which ANTIK will be integrated.

References

[1] Peter F. Brown and Stephen A. Della-Pietra. 1993. The Mathematics of Statistical Machine Translation. *Computational Linguistics*, 19 (2): 263-313.

[2] James W. Carey. 1989. *Communication as Culture*. Unwin Hyman, Boston, MA.

[3] Noam Chomsky. 1957. *Syntactic Structures*. Mouton & Co., The Hague.

[4] D. O. Hebb. 1949. *The Organization of Behavior.* John Wiley & Sons, Inc., New York, NY.

[5] Lucian Leahu, Phoebe Sengers, and Michael Mateas. Interactionist AI and the promise of ubicomp. *UbiComp'08*, September 21-24, 2008, Seoul, Korea.

[6] B. Malinowski. 1923 The Problem of Meaning in Primitive Languages. In *The Meaning of Meaning*, Charles K. Ogden and Ian A. Richards (Eds.), Routledge, London, UK, 146-152.

[7] Marvin Minsky, 1968. *Semantic Information Processing*. MIT Press, Cambridge, MA.

[8] A. Newell, J. C. Shaw, H. Simon. 1957. Elements of a Theory of Human Problem Solving. RAND Corp, Santa Monica, CA.

[9] B. F. Skinner. 1957. *Verbal Behavior*. Appleton-Century-Crofts, New York, NY.