# Analogical Question Answering in a Multimodal Information Kiosk

**Jason R. Wilson** JRW@NORTHWESTERN.EDU
**Kezhen Chen** KEZHENCHEN2021@U.NORTHWESTERN.EDU
**Maxwell Crouse** MVCROUSE@U.NORTHWESTERN.EDU
**Constantine Nakos** CNAKOS@U.NORTHWESTERN.EDU
**Danilo Neves Ribeiro** DNRIBEIRO@U.NORTHWESTERN.EDU
**Irina Rabkina** IRABKINA@U.NORTHWESTERN.EDU
**Kenneth D. Forbus** FORBUS@NORTHWESTERN.EDU
Computer Science, Northwestern University, Evanston, IL 60208 USA

## Abstract

While many have developed sophisticated information kiosks, few have used a kiosk as a platform to investigate multimodal question answering. A kiosk provides a rich platform for multimodal inputs and outputs while capturing user inquiries in-the-wild. We describe here a deployed information kiosk that generates responses using analogical question answering. This data-efficient learning mechanism allows the system to answer a broad set of questions while being trained on only a few training example questions. The system is deployed, and we report its usage over 151 days for which we collected data. During this time, we identified questions that were not being answered, then updated the training of the analogical questions answering with four new examples, enabling the system to answer a whole new class of questions. This shows the utility of our data-efficient learning approach to question answering while also demonstrating the value of the kiosk as a platform for investigating question answering in-the-wild.

## 1. Introduction

We are exploring multimodal question answering in a naturalistic environment. As a first step in this effort we have developed and deployed a multimodal information kiosk designed to provide information to students, faculty, staff, and visitors to the Computer Science Department at Northwestern University. Having moved into a new office space just weeks before the system was deployed, we recognized that there would be a need for people to learn their way around the new location. Additionally, with the new school year starting around the same time, many would also need information regarding courses, office hours, and academic advising. To meet some of the needs of the people as we entered the new space, we developed a multimodal information kiosk powered by an analogical question answering system.

Developing a multimodal information kiosk is not a new idea, though most previous efforts have addressed how to improve the interaction through the use of multimodal sensor capabilities, interactive displays, and virtual agents or robots. For example, MACK (Cassell et al., 2002; Stocky & Cassell, 2002) and NUMACK (Hasegawa et al., 2010) used an embodied conversational agent to give directions using a combination of visual processing, spatial reasoning, and gestures to direct people to a requested location. MIKI was also designed to assist students, faculty, and visitors of an academic building detected when users were present and processed spoken input to produce verbalized answers conveyed by an avatar along with videos, 3-D maps or personnel page (McCauley & D'Mello, 2006). In addition to using avatars, some have used robots, such as the Roboceptionist at CMU (Lee et al., 2010), for which they examined sociability and politeness by analyzing natural language, or the Directions Robot at MSR (Bohus et al., 2014), which provided gestures (similar to MACK) and natural language instructions for how to find a location. Similar to MIKI, the Directions Robot also used speech recognition and vision components to recognize when a person was engaging with the robot.

In contrast to these previous efforts, we leveraged many off-the-shelf capabilities to manage the interaction and instead provide a novel approach to question answering in a naturalistic environment. We describe here *analogical question answering*, which can be trained to answer a broad set of questions with only a few training examples for each type of question. Combined with a knowledge base that includes commonsense facts and knowledge specific to the Computer Science Department at Northwestern University, analogical question answering enables the information kiosk to assist users in finding locations and retrieving information regarding faculty, courses, and events. Extending our prior work on analogical question answering, one contribution of this work is that the question answering system provides multimodal responses by generating commands (e.g., to display a location on a map) to the UI in addition to textual responses. The kiosk has been successfully deployed for over six months, during which we have collected data on how users interact with the kiosk and the types of questions users have. Over the course of the six months, we have been able to use the collected data to extend the question answering system to include new types of questions people asked the kiosk. The kiosk has been shown to be a useful tool for collecting data on real usage, which provides a new type of data for investigating question answering. The remainder of this paper reviews question answering approaches, describes the architecture of the information kiosk, and discusses the data we have gathered from the deployed system.

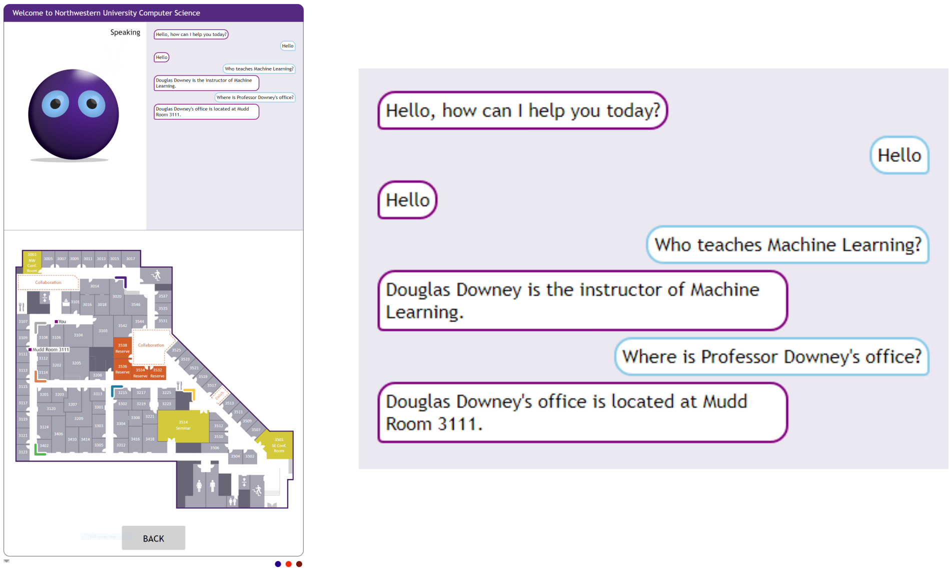
## 2. Background

Many question-answering systems operating over structured knowledge learn to map directly from natural language inputs to domain specific logical forms (Zelle & Mooney, 1996; Liang, Jordan, & Klein, 2013, Kwiatkowski et al., 2010). While there are some benefits to this methodology, e.g. the system designer needs only to consider the inputs and outputs and can avoid tinkering with any internal representations, it comes with a number of drawbacks. First, such systems generally need to be retrained from scratch for every new domain to which they are applied (Berant et al, 2013; Liang, Jordan, & Klein, 2013). Second, such methods are often data intensive, requiring hundreds (Ge, Ruifang, 2010), if not thousands (Berant et al., 2013; Tafjord et al., 2018), of examples to achieve acceptable performance. And while methods have been devised to mitigate the cost of acquiring such large amounts of training data, e.g. training from natural-language-only question-answer pairs (Berant et al., 2013; Liang, Jordan, & Klein, 2013) or reverse-engineering queries from crowdsourced questions (Tafjord et al., 2018), they are not universally applicable to every question-answering domain, e.g. for the multi-modal (speech and visual) output required in the kiosk question-answering domain. An extra challenge is that the kiosk has to deal with varied questions asked by real-world users, instead of templated or domain specific questions. Open-ended question-answering is still a challenge for AI systems, where state-of-the-art results are still far from human performance (Clark et al., 2018, He et al. 2018).

The kiosk uses Analogical Question-Answering (AQA) (Crouse, McFate, & Forbus, 2018a, 2018b) to answer posed questions. The decision to utilize AQA was prompted by the following considerations: (1) AQA is data efficient and was shown to produce results comparable to state-of-the-art with an order of magnitude less data on the Geoquery dataset (Crouse, McFate, & Forbus, 2018a). (2) AQA is easily extensible, needing only a handful of examples to handle a new type of question. (3) this system is efficient in terms of computation requirements, being able to run on a single desktop computer without the need of more expensive components such as GPUs. (4) AQA can adapt to longer questions. In (Crouse, McFate, & Forbus, 2018b), AQA was shown to be extendable to scenario questions involving multiple sentences (often requiring coreference resolution). For interactions with the kiosk, we had initially suspected that users would pose follow-up questions that required the processing of longer inputs and the reinterpretation of prior utterances (however, this was not an accurate assumption). While AQA has been described in (Crouse, McFate, & Forbus, 2018a, 2018b), we briefly provide an overview of how it is used in the kiosk in section 3.2.

## 3. System Architecture

The information kiosk interacts with a user through natural language to provide locations of people's offices, the kitchen, and bathrooms; course information; and information on research groups. The system integrates a touch screen, a 3d depth camera with a microphone array, an existing platform for multimodal interaction, and a cognitive architecture for higher-level reasoning. An example interaction with the kiosk is presented in Figure 1, and a video is available at https://youtu.be/aAs7w6OG94Q.



Hello, how can I help you today?

Hello

Hello

Who teaches Machine Learning?

Douglas Downey is the instructor of Machine Learning.

Where is Professor Downey's office?

Douglas Downey's office is located at Mudd Room 3111.

Figure 1: On the left is the full view of the kiosk screen, and on the right is a close-up of the chat window.

The system is organized into three layers: Hardware, PsiKi, and Companion (see Figure 2). The Hardware layer consists of a touch screen with built-in speakers and a Kinect, for cameras and microphone array. The PsiKi layer handles lower-level reasoning by processing data received from the hardware, sending data to the hardware, and communicating with a Companion. The top layer is a Companion (Forbus & Hinrichs, 2017), which does the higher-level reasoning using natural language understanding and generation and analogical question answering. PsiKi and Companion are described further next.

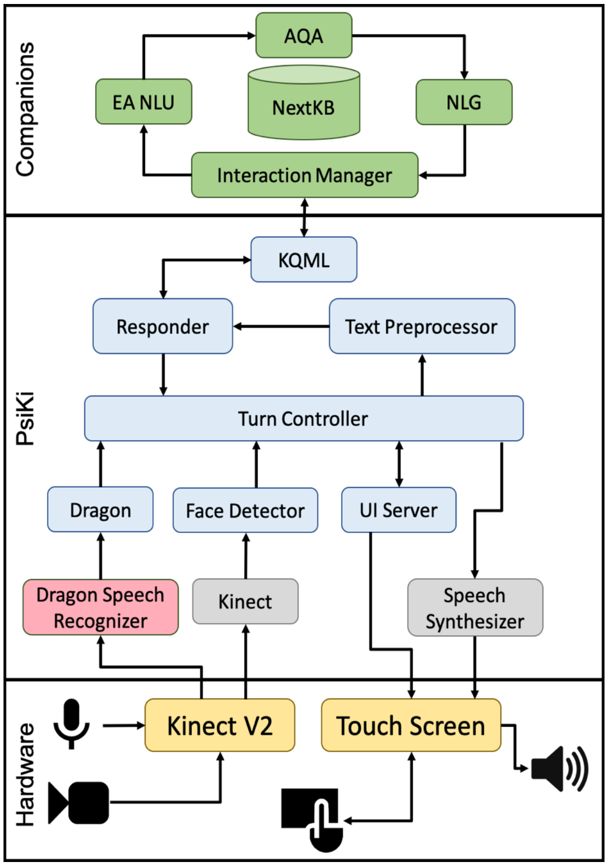


Figure 2: Kiosk Architecture with Companion components (green), new \psi components (blue), existing \psi components (gray), commercial software (pink), and hardware (yellow).

### 3.1 PsiKi

The PsiKi subsystem of the kiosk is responsible for handling perception and lower-level reasoning by managing the flow of streaming data from sensors, communications with the GUI, communications with the Companion, turn-taking, and some reactive responses to user input.

PsiKi uses Microsoft's Platform for Situated Intelligence (\psi)[1] to integrate different inputs and outputs and synchronize them to work in a single pipeline. \psi is an open, extensible framework that enables the development, fielding and study of situated, integrative-AI systems (Bohus et al., 2017). The \psi system operates over streaming data to support acting in the real world with low-latency and under uncertainty. Any AI technologies and systems that operate over streaming data can be integrated by \psi to build multimodal applications. The \psi system provides a runtime environment and a collection of components to process and produce streams of data. We extended this, introducing additional components to create our PsiKi system. The middle of Figure 2 shows the PsiKi architecture, and we describe next the key components of the PsiKi system.

### *3.1.1 Sensory Inputs*

We want users to be able to speak to the kiosk in natural language. Additionally, since the kiosk is in a high traffic area, we want the kiosk to recognize only speech directed at the kiosk and not pick up other conversations in the area. To determine if someone is attempting to use the kiosk, the Face Detector component uses video and skeleton data from the Kinect to detect faces. If a face can be detected, then the person is likely to be looking at the kiosk and could be a potential user. Once a face is detected, the Speech Recognizer is enabled, allowing incoming audio to be processed. We wrapped the Dragon Naturally Speaking dictation engine as a \psi component and integrated it into the pipeline. The dictation engine allows the use of natural language that is not constrained, as it would be with a hand-tuned grammar. When the Speech Recognizer produces an utterance, the \psi pipeline sends it to the Turn Controller for further processing.

### *3.1.2 Turn-Taking*

The kiosk is designed to allow the user and the kiosk to take turns in communicating. To manage this turn-taking, the Turn Controller determines when to transition between five states: Sleeping, Listening, Waiting, Thinking, and Speaking. The Sleeping state is for when no one is using the kiosk. The system is awoken when either a face is recognized (enters the Listening state) or the touch screen is used (enters the Waiting state). Once the user has completed an input, either by speaking or through the touch screen, the Turn Controller enters the Thinking state. When a response is ready, the Turn Controller transitions to the Speaking state, and when speaking is completed returns to either the Listening or Waiting state, depending on how the system was awoken. For each of the states, the UI textually displays the state and the avatar has a corresponding animation.

During the Thinking state, the Responder is responsible for generating a response to the user by either generating an automatic response or forwarding the input to the Companion to process. A small set of inputs create reactive responses by the Responder, which uses simple keyword matching to identify to which inputs to automatically respond. The reactive responses include greetings (e.g., "Hello"), inquiries about the bathroom or office hours, and requests for bus times. For inputs that require more reasoning, the Responder passes the input along to the Companion. To communicate with the Companion, PsiKi uses the Knowledge Query and Manipulation Language (KQML) (Finin et al., 1994), which is designed to allow knowledge-based agents to communicate at the knowledge level (as opposed to lower-level streaming data, as \psi does). We

[1] http://microsoft.github.io/psi

use KQML to provide a communication layer between the low-level processing of PsiKi and the higher-level reasoning of Companion, since the agents inside a Companion also use KQML.

### *3.1.3 User Interface*

The User Interface presents information to the user via a touch screen and allows the user to provide input. The graphical interface provides an avatar, a chat window, and an input/output panel. The avatar is a simple purple ball with big eyes and no mouth – eliminating the need to attempt synchronization of the mouth and phonemes. The chat window, similar to one found in texting applications, allows the user to see what the kiosk has heard and what the kiosk is saying. The bottom half of the screen toggles between being used for input and output. Initially, the screen displays input options, providing a keyboard and convenience buttons for question templates (e.g., "Where is ___?") and common words (e.g., "Professor" or "Room"). For output, this portion of the screen may display a map or a calendar.

To control the UI, PsiKi supports three commands: psikiSayText, psikiShowMap, and psikiShowCalendar. The psikiSayText command adds the given text to the chat window. The psikiShowMap command results in the UI showing a map of the floor of the building along with a label for a specific location. For example, if the kiosk is asked the location of an office, the kiosk will display the map and mark the location on the map with the requested location. The psikiShowCalendar command causes the UI to show a calendar of the current day.

In addition to presenting information visually, the kiosk also uses a speech synthesizer to present information orally. The text displayed in the chat window of the UI is also spoken. Once the kiosk has completed speaking, the Dialogue Manager returns to a state in which it will accept more user input.

## 3.2 Companion

Higher-level reasoning for the kiosk is handled by a Companion (Forbus & Hinrichs, 2017), a cognitive system built on a distributed agent architecture. PsiKi is registered with the Companion as an agent, allowing it to pass KQML messages to the Interaction Manager, the agent responsible for natural language interaction. The Interaction Manager receives a user utterance from PsiKi and parses it using EA NLU (Tomai & Forbus, 2009) to produce a semantic interpretation. This interpretation is then passed to Analogical Question Answering (AQA), which constructs an appropriate knowledge base query to answer the user's question. Once the answer has been retrieved from the KB, a Natural Language Generation (NLG) component translates it into English text. Finally, the Interaction Manager sends the output text back to PsiKi, along with any additional instructions regarding how to present it or what else to display.

### *3.2.1 EA NLU*

EA NLU (Tomai & Forbus, 2009) performs natural language understanding on the user utterance. EA NLU is a semantic parser built on Allen's (1994) bottom-up chart parser. The parser uses a head-driven, feature-based grammar and the NULEX lexicon (McFate & Forbus 2011) to produce a parse tree. Along the way, it builds up a semantic interpretation for the utterance using compositional frame semantics derived from FrameNet (Fillmore, Wooters, & Baker 2001) and mappings from English words to concepts in the knowledge base. The interpretation is represented in predicate calculus using the NextKB ontology (Forbus & Hinrichs, 2017). EA NLU explicitly represents ambiguity in the form of choice sets which are passed along for later processing. The interpretation produced by EA NLU is used as input to AQA.

### *3.2.2 Analogical Question Answering*

Analogical Question Answering (AQA) operates in two modes: training and execution (see Figure 3). The semantic choices, which are output from EA NLU, represent the natural language question and are inputs during both modes. The logical forms are a predicate calculus representation of the intended response to the question and are input during training and an output during execution. Previous AQA systems used natural language answers as one of the inputs to the system. In this task, we use logical forms because responses may include commands to the UI, e.g., displaying a location on a map. The query cases contain the selected semantics paired with the logical forms. Training of AQA outputs a small set of query cases, and the full set of query cases are used as inputs during execution. A full description of AQA is available in (Crouse, McFate, & Forbus, 2018a), and a brief overview of the Training and Execution modes follow.

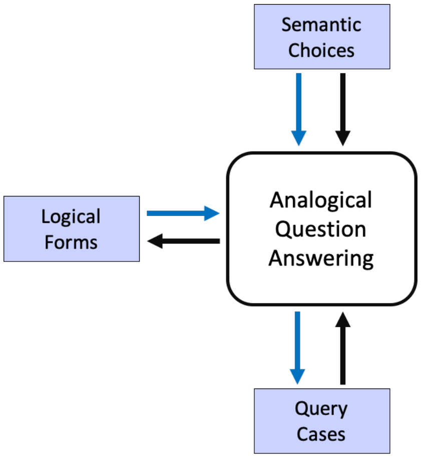


Figure 3: Analogical Question Answering inputs and outputs during training (blue arrows) and execution (black arrows).

**Training:** The objective of AQA during training is to define a query case that maps the semantic choices to the logical forms by determining the set of semantic choices needed to justify the given logical form. By analogy with Horn clauses, the objective of AQA is to determine the antecedents (i.e., semantic choices) that will be used to justify a consequent (i.e., a logical form). As inputs, AQA training is given a set of semantic choices and an associated domain-specific logical form and generates query cases, which bridge between the semantic parser outputs and the domain-specific logical form. To demonstrate how AQA operates during training, we consider the natural language input, “Can you tell me what Doug Downey’s email is?” From this input, EA produces a semantic parse (see Figure 5), which is then paired with a logical form (see Figure 4). For this question, the associated logical form is a generic query that finds the email address of the desired Northwestern University-affiliated person and returns that address to the user.

```
From "tell"
   - (isa tell51698 StatingSomething)
From "tell me ... email"
   - (infoTransferred tell51698 email51832)
   - (fe_proposition tell51698 email51832)
From "Doug Downey"
   - (isa DouglasDowney NUPerson)
From "Doug Downey's email"
   - (possessiveRelation DouglasDowney email51832)
From "email"
  - (isa email51832 EMailMessage)
```

Figure 5: Semantic parse of “Can you tell me what Doug Downey’s email is?”

```
(and (emailOf person123 address123)
     (isa person123 NUPerson)
     (isa address123 StringObject))
```

Figure 4: Logical form that is paired with “Can you tell me what Doug Downey’s email is?”

The first step in AQA training is to find a minimal set of non-conflicting semantic choices, which AQA does using a bipartite matching algorithm followed by building a Steiner-tree. First, a one-to-one bipartite matching algorithm finds a correspondence between expressions in the set of semantic choices (Figure 4) and expressions in the logical form (Figure 5). While finding correspondences, the matching algorithm optimizes for ontological relatedness and structural similarity while also ensuring no two semantic choices used in the match conflict with one another. The ontological relatedness of two expressions is given by the ways one could connect their constituent elements through the knowledge base. For example, the semantic choice `(isa email51832 EMailMessage)` is connected to the logical form expression `(emailOf person123 address123)` via the concept `EMailAddress`. The structural similarity is measured by the number of times neighboring expressions in the semantic choices are matched with neighboring expressions in the logical forms. For example, `(isa DouglasDowney NUPerson)` and `(isa email51832 EMailMessage)` are neighbors, which are linked through the expression `(possessiveRelation DouglasDowney email51832)`, corresponds with

```
(queryCase
   (and(emailOf DouglasDowney address123)       - person123 -> DouglasDowney
       (isa DouglasDowney NUPerson)             - person123 -> DouglasDowney
       (isa address123 StringObject))
   (and(isa email51832 EMailMessage)                       – From matching
       (isa DouglasDowney NUPerson))                       – From matching
   (and(possessiveRelation DouglasDowney email51832)) – From Steiner-tree
```

Figure 6: Query case for "Can you tell me what Doug Downey’s email is?"

the pair of logical form expressions `(emailOf person123 address123)` and `(isa person123 NUPerson)`, which share an entity. Since these pairs of neighbors correspond, the structural similarity score is incremented, ultimately contributing to their pairs being included in the final query case (see Figure 6).

The matching may result in disconnected expressions, where there is no structure tying together the entities in the semantic choices. For example, there is no connection between `email51832` and `DouglasDowney`. To make this connection, a Steiner-tree algorithm selects the smallest (also non-conflicting) set of semantic choices that connects each of the variables. Once the Steiner-tree algorithm completes, AQA substitutes variables in the logical form using values from the semantic choices. For example, all instances `person123` are replaced with `DouglasDowney`. The query case is then complete and stored away for retrieval during execution.

**Execution:** AQA is given a set of semantic choices representing a natural language question, and it generates a logical form, from which the Companion produces a natural language response and provides commands to PsiKi to display additional information. To generate the logical form, it takes semantic choices from EA and uses MAC/FAC (Forbus et al., 1995) to do analogical retrieval of a small number of the query cases generated during training. It then uses SME (Forbus et al., 2017) to instantiate the query cases with the particular entities and variables from the input question's semantic choices. With a set of query cases that have been instantiated for the input question, the algorithm presented in (Crouse, McFate, & Forbus, 2018a) selects the smallest set of query cases whose associated semantic choices cover the entirety of the input question while maintaining that no two query cases can be selected if their activation conditions contain conflicting semantic choices. The consequents of selected query cases are conjoined to produce a query form.

### *3.2.3 NLG*

Once AQA has produced a query form, the KB is queried using the query form to determine the answer to present to the user. For example, part of the query produced for "Where is Professor Forbus' office?" is `(officeLocation KenForbus ?office123)`, which matches the fact `(officeLocation KenForbus "Mudd Room 3113")`from the KB. The response is passed through a template-based NLG system to convert it into an English string, in this case "Ken Forbus' office is located at Mudd Room 3113." The response is then packed in a KQML message to send to PsiKi, along with any display commands found in the query.

By default, the entire query is converted into English except for `isa` statements and PsiKi commands. However, the forms useful for knowledge retrieval are not always useful to present to the user. When this mismatch occurs, the PsiKi command `psikiSayText` is used to specify the appropriate expression to verbalize. For example, `(psikiSayText (courseTimeString <course> <time>))` will yield an output string that says the time of the course, regardless of the other expressions used in the query.

## 4. Deployment

### 4.1 Pre-deployment Survey

Before deploying the kiosk, we surveyed the Computer Science students to get a sense of what types of questions they would like to be able to ask the kiosk. An email announcement with a link to a survey was sent to all CS students (undergraduate and graduate). We received a total of 42 responses to our pre-deployment survey. Each response contained three questions a user would want to be able to ask the kiosk and optional comments. The questions were categorized by type and were used as inspiration for training the kiosk's question answering system.

Our primary focus was on the three most common question types: Office Location, Resource Location, and Office Hours (Table 1). Within and across the three categories, proposed questions varied in both the grammatical structure of the question (e.g., "Where is X?" vs. "How do I find X's office?") and the referring expression (e.g., Professor {LastName} vs. {FirstName} only). The appropriate response differed, too. For example, some questions could be answered verbally, but others required showing a map or calendar. See Section 3.1 for description on how such variations are handled by the system.

Table 1: Pre-deployment survey of questions to ask the kiosk

| Question Categories | # | Example | Handled by Current Version |
|---|---|---|---|
| Office Location | 26 | Where is professor {name}'s office? | Yes |
| Resource Location | 21 | Where is the bathroom? | Yes |
| Office Hours | 17 | When are {name}'s office hours? | Yes[2] |
| Professor Availability | 11 | Is {name} available right now? | No |
| Course Information | 10 | What courses are available next quarter? | Partial |
| Study Room Booking | 6 | Can I book {room} for {time/day}? | No |
| Administrative Tasks | 5 | Where do I get a major declaration form? | No |
| Library Tasks | 3 | Is {book} available for checkout in the library? | No |
| Not Serious | 16 | What is the meaning of life? | Some Easter eggs |
| Miscellaneous/Other | 10 | What events are scheduled for today? | No |

### 4.2 Setup and Deployment

The information kiosk is deployed in the new space for the Computer Science department at Northwestern University. With a new space, there can be ample confusion on where to go, thus allowing the kiosk to help fill an immediate need.  The intent is for the kiosk to provide information to students, faculty, and visitors. The kiosk is mounted directly to the wall (using a VESA mount) at high traffic area (near the reception desk and main elevators) in the Computer Science space.. As there are a large number of people passing the kiosk, it is designed to avoid

[2] Note that, while the system is capable of answering questions about office hours, administrative hurdles have caused that information to be incomplete and/or out of date.

incorrect activation, such as people casually passing the kiosk or having a conversation in its vicinity. Since the area may be too noisy for speech interaction, a user may also wake up the kiosk by touching the screen and then continuing to provide touch input. Once touch input has been initiated, speech is disabled.

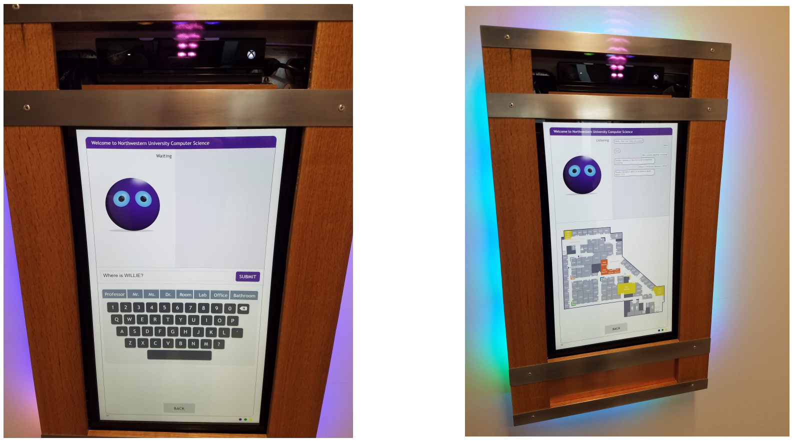


Figure 7: Kiosk is in a cabinet and mounted on the wall. A Kinect is mounted above the touchscreen, and a ring of LEDs surround the cabinet.

The information kiosk, as seen in Figure 7, consists of a monitor and a Kinect V2 sensor, which are connected to a PC with the kiosk programs running in the lab behind the wall. The monitor is a 27” Viewsonic TD2740 touchscreen with 1920x1080 resolution, 10 touch points, decently tough glass, and 178 degree viewing angle. The PC computer has an i7 intel processor, 64GB RAM, 1TB SSD primary drive, and a 3TB data hard drive. The Kinect V2 provides both the visual information and audio information from the environment. On the vision side, the Kinect V2 is capable of 1080p color video, active infrared sensing, depth sensing, body tracking, facial tracking, gaze detection etc. On the audio side, the Kinect V2 can provide a high-quality audio signal. Both the visual information and audio information of Kinect V2 sensor are sent to PsiKi for people detection and speech recognition. As shown in Figure 7, the monitor is placed at the center of cabinet, between compartments for sensors (the top and bottom compartments are each 19’’ wide x 5’’ high x 3.5’’ deep). The Kinect is installed in the top compartment, and the bottom compartment is currently vacant. The cabinet face is removable to access the hardware and the face is designed to be locked for security. There are LED strip along the back of the front panel providing halo lighting to attract people’s attention.

### 4.3 Usage Statistics

The kiosk has been operating for over six months, over which we have collected data from 151 days of usage. After filtering for test, incomplete, and unintelligible questions, we saw a total of 538 questions over the 6-month deployment period. Of these, the system answered 233 (see Table 2, which has the same rows as the pre-deployment survey results in Table 1). Figure 9 shows considerable usage when the system was first deployed and then regularly having 10-20 inputs each week. Weeks 11,12, 13, and 23 are school breaks, and the kiosk had minimal usage during this time. The first week after the winter break had a small surge usage, and the spikes in usage during weeks 19 and 21 were likely the result of special events (with many guests) hosted by the Computer Science department.

Table 2: Categories of questions asked of the deployed kiosks

| Question Category | Total Qs | Total answers | Unique Qs | Unique answers | Example |
|---|---|---|---|---|---|
| Office Location | 232 | 95 | 144 | 41 | Where is {name}? |
| Resource Location | 161 | 76 | 75 | 17 | Where is the bathroom? |
| Office Hours | 7 | 7 | 7 | 7 | When is {name}'s office hour? |
| Professor Availability | 1 | 0 | 1 | 0 | What is {name}'s schedule? |
| Course Information | 27 | 2 | 20 | 1 | When is {class}? |
| Study Room Booking | 0 | 0 | 0 | 0 | N/A |
| Administrative Tasks | 3 | 0 | 2 | 0 | Who is director of graduate studies? |
| Library Tasks | 0 | 0 | 0 | 0 | N/A |
| Not Serious | 45 | 13 | 31 | 1 | Are you a boy or a girl? |
| Miscellaneous/Other | 62 | 40 | 32 | 10 | When is the next {bus number}? |
| Total | 538 | 233 | 312 | 77 | |

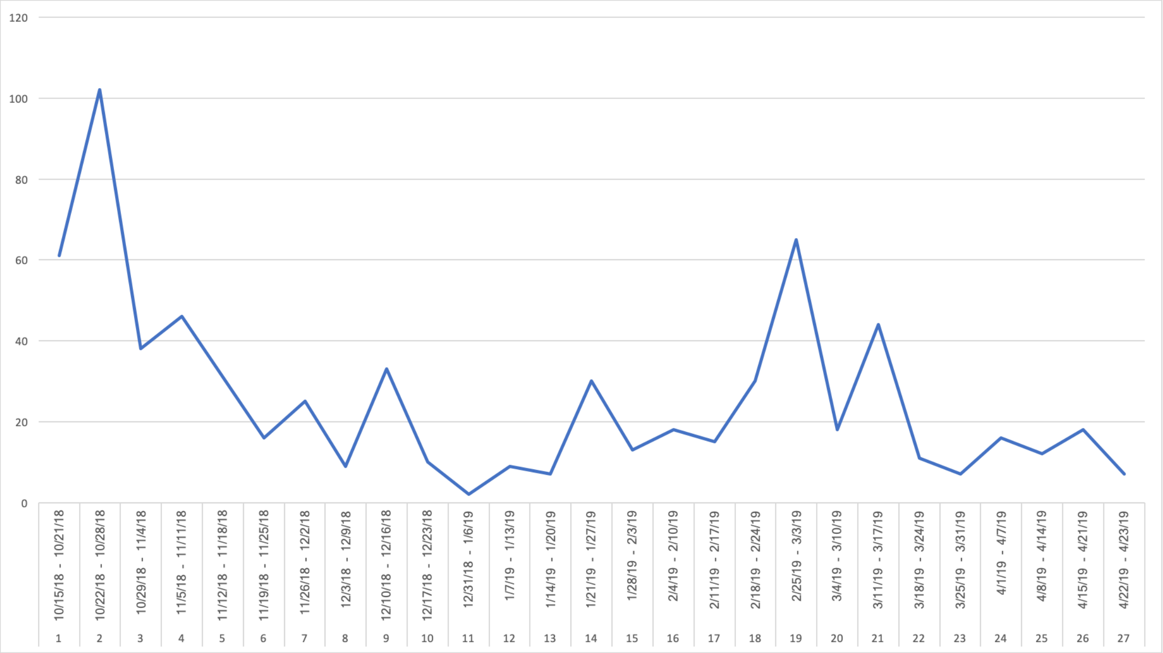


Figure 8: Number of questions asked of the kiosk, week-by-week.

Question patterns were similar to those in the pre-deployment survey, with most questions falling into the Office Location and Resource Location categories. Questions in the Study Room Booking and Library Tasks categories were never asked, while those in the Professor Availability and Administrative Tasks categories were extremely rare (one and three total, respectively). On the other hand, the Miscellaneous category — a catch-all for question types that did not fall into the categories from the pre-deployment survey—saw a total of 62 questions, of which the kiosk answered 40.

The plurality of the questions that fell into the Miscellaneous category (17) were about the bus schedule. Because we anticipated this pattern, the kiosk answered all of them correctly. Other miscellaneous questions included questions about research groups (14), the kiosk itself (6), on-campus events (4), real-world information (e.g., “What time is it?”, “What is the population of Chicago?”; 9), and professor contact details (2). The remaining questions in this category were variations on, “Show me the map”—an extension to the Office Location and Resource Location categories.

### 4.4 Training

At present, the AQA system is trained to handle nine question types with 102 training examples. Table 3 lists each of the question types along with one of the training examples and the number of training examples used.

Table 3: Training question types

| | Question Type | Information Category | Example | No. Training examples |
|---|---|---|---|---|
| 1 | Office Location | People | “Where is Professor Forbus's office?” | 14 |
| 2 | Courses Taught | People | “What does Professor Sood teach? | 6 |
| 3 | Email Address | People | “What is Douglas Downey's email?” | 21 |
| 4 | Webpage | People | “What is Jason Wilson's webpage?” | 8 |
| 5 | Room Location | Building | “Where is room 3011?” | 5 |
| 6 | Members List | Group | “Which faculty are in Algorithms?” | 9 |
| 7 | Membership | Group | “Which group is Anindya De in?” | 5 |
| 8 | Instructor | Course | “Who is teaching CS 371?” | 24 |
| 9 | Time | Course | “When is CS 371?” | 10 |

Different training questions reflect the expected variation in how a type of question is posed. For instance, in the Office Location category, AQA is trained with both “Where is Professor Forbus’s office?” as well as “How can I find Professor Forbus’s office?”. Some questions have more phrasings that can be used to ask the question, resulting in more training examples. Fewer training examples may be possible in some cases, but we have not done any tests on how many examples are necessary to capture a given set of inputs.

To train AQA on all 102 questions, we us a standard desktop machine (Dell OptiPlex with i7-8700 CPU @ 3.20GHz and 64 GB RAM running Windows 10 Pro) and are able to complete the training in approximately four minutes.

### 4.5 Examples

The following is a set of actual questions asked of the kiosk, along with the responses given to the user. In many cases, a “Sorry, I don’t understand” response was generated, indicating that the kiosk was unable to answer the question.

- "Professor Downey's office" >> "Douglas Downey's office is located at Mudd Room 3111."
- "Downey's office" >> "Sorry, I don't understand."
- "Where is Ian?" >> “Sorry, I don’t understand”
- "Where is Ian Horswill?" >> "Ian Horswill's office is located at Mudd Room 3537."
- "Where is room 3014?" >> “Sorry, I don’t understand.”
- "Where is Delta Lab?" >> "Sorry, I don't understand."
- "Where is Tech?" >> "Sorry, I don't understand."

Many of the errors can be attributed to gaps in language coverage. For instance, in the second question shown above, the link between the word “Downey” and the concept for Doug Downey in the knowledge base was not defined (while the link between “Professor Downey” and Doug Downey was). Similarly, for the third question, “Ian” is not linked to the concept for Ian Horswill. At first glance, it may seem that resolving this issue is simply a matter of adding a link between first and last names to the appropriate knowledge base concepts. However, doing so could introduce a new problem stemming from ambiguity.

The question “Where is room 3014?” is an instance of a simple question that was not expected when the kiosk was first trained. As such, there was neither a link between “room 3014” and the corresponding knowledge base concept nor a training example that would allow the system to answer such questions. To extend the kiosk, four new training examples were added to the overall set of test questions to cover the various ways the above question might be posed, e.g. “Can you tell me where room 3014 is?”.

In the last two questions, “Tech” and “Delta Lab” represent gaps in the knowledge base. Names of nearby buildings (and their nicknames) have not been introduced to the knowledge base yet. Similarly, names (and other related information) of labs affiliated with the Computer Science department are not currently in the knowledge base.

## 5. Discussion

We have successfully deployed a multimodal information kiosk that students, faculty, staff, and visitors have been able to use to get information about the Computer Science department and the new space it has recently occupied. To accomplish this, we have integrated PsiKi, which handles perceptional and lower-level reasoning, with a Companion, which handles higher-level reasoning. Given a user’s question, the Companion uses analogical question answering to construct an appropriate query to a knowledge base, and from the result of the query we are able to generate an appropriate response. The response includes textual output to be displayed and spoken by PsiKi, and it may also contain additional outputs, such as a command to display a location on a map.

Overall, we see the success of the deployed kiosk not simply as its ability to provide information but as a platform by which we can continue to investigate multimodal question answering in-the-wild. While users got responses to 233 questions over 151 days of data collection, more importantly, we have analyzed the questions to identify 312 unique questions that people actually asked the kiosk, providing us with a dataset by which we can continue to evolve the analogical question answering system. As a demonstration of the utility of the dataset, we identified that questions about the location of a room given the room number did not generate a response. and added four new training examples to the analogical question answering system to enable it to generate appropriate responses for these types of questions. We are continuing to work on streamlining the process of identifying unanswered questions, providing a small set of training examples, and updating the system to answer the additional questions.

## 5.1 Challenges

We faced many challenges in developing the kiosk, and three of the most prominent ones consisted of difficulties in acquiring data, detecting user engagement, and accurate speech recognition. Some of the difficulties in acquiring data were related to gathering information from unreliable sources. Some of the information about faculty and courses was scraped from corresponding web pages, but the information was often out of date. All of the faculty were in the process of changing offices, and course information (i.e., instructor, terms offered, time and place) changed irregularly but not infrequently. Additionally, course information on web pages was too often stale. In fact, stale information was regularly the challenge to integrating new data. For example, we experimented with displaying a calendar of office hours, where the calendar was an aggregation of calendars that instructors setup for their own courses. After the initial effort to collect office hour information, the information was not updated, eventually leading to an empty calendar of office hours, which can be misleading. Future efforts in regards to office hours, and all knowledge really, must ensure that there will consistently be a reliable source of knowledge. We believe that committing to including knowledge about a particular topic requires committing to maintaining the freshness of the knowledge.

In addition to knowledge acquisition, we faced some challenges in implementing the vision and speech capabilities of the kiosk. On the vision side, we check for user engagement by detecting a person's face using a Kinect. However, there are many different situations where face detection did not perform well, and many of these cases are related to the hardware design. The angle of the sensor often made it such that it could detect either a shorter person or a taller person but not both. Additionally, the Kinect is designed to work best when a person is at least three feet away, causing the face detection to typically fail when a person is close to the screen, as one may be when about to touch the screen. An unfortunate consequence of this was the kiosk going to sleep during interactions because the face detection fails to track the face at a close distance. To handle the specific case of a person using the touch screen, we disabled face detection (and its associated behaviors) once a person has touched the screen.

On the speech side, the speech recognizer initially performed badly on many names, especially non-Anglican names. To increase the scope of the speech recognition, we added faculty names to a dictionary of additional terms for the Dragon speech engine to recognize. We also needed to perform some processing on the output of the speech recognizer to modify some number usage (especially course and room numbers) and to filter out some utterances (e.g., most one or two word utterances, which often were incomplete utterances or disfluencies).

### 5.2 Future Work

In the long-term, we envision the kiosk to be a fully conversational agent. This means going beyond answering factoid-based questions (at it does now) to holding entire conversations and providing useful information based on what it knows about its interlocutor, without being asked directly. To this end, we are developing a Friends of the Kiosk program, where users can opt in to allow the kiosk to learn about their interests and preferences. This will allow the kiosk to make recommendations and suggestions to those users. Furthermore, through analogical generalization, the kiosk will be able to extend these inferences to users who have not agreed to the use of their own data. For example, if the kiosk has learned that all of the graduate students who have opted in to the Friends program like events with free food, it might recommend such an event to another graduate student—even if it has no particular knowledge of her preferences.

## 6. Conclusion

We described a multimodal information kiosk using analogical question answering that has been successfully deployed for over six months. The kiosk is triggered by the presence of a user and allows users to provide input via speech or touch. For most user inputs, the kiosk uses the Companion cognitive architecture to do analogical question answering and natural language understanding and generation. The deployed system has captured hundreds of user interactions, allowing us to collect in-the-wild data on the types of questions users ask the kiosk. Using the collected data, we are able to easily update the kiosk, providing only a few additional training examples to enable the kiosk to answer an additional type of question. Overall, the kiosk provides a platform by which we can continue to investigate question answering in-the-wild.

## Acknowledgements

We thank Dan Bohus and others at Microsoft Research for early access to \psi and their assistance in working with the platform. We also thank the students in the Conversational Interfaces course, which contributed to some of the early stages of development of the kiosk.

## References

Allen, J. F. (1994). *Natural Language Understanding*. (2nd ed). Redwood City, CA: Benjamin/Cummings.

Berant, J., Chou, A., Frostig, R., & Liang, P. (2013). Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1533-1544).

Bohus, D., Andrist, S., & Jalobeanu, M. (2017, November). Rapid development of multimodal interactive systems: a demonstration of platform for situated intelligence. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction* (pp. 493-494). ACM.

Bohus, D., Saw, C. W., & Horvitz, E. (2014, May). Directions robot: in-the-wild experiences and lessons learned. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems* (pp. 637-644).

Cassell, J., Stocky, T., Bickmore, T., Gao, Y., Nakano, Y., Ryokai, K., ... & Vilhjálmsson, H. (2002, February). Mack: Media lab autonomous conversational kiosk. In *Proc. of Imagina* (Vol. 2, pp. 12-15).

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., & Tafjord, O. (2018). Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Crouse, M., McFate, C., & Forbus, K. (2018a, April). Learning from Unannotated QA Pairs to Analogically Disambiguate and Answer Questions. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Crouse, M., McFate, C., & Forbus, K. (2018b). Learning to build qualitative scenario models from natural language. In *Proc. 31st Int. Workshop on Qualitative Reasoning (QR'18)*.

Fillmore, C. J., Wooters, C., & Baker, C. F. (2001). Building a large lexical databank which provides deep semantics. In *Proceedings of the 15th Pacific Asia Conference on Language, Information and Computation* (pp. 3-26).

Finin, T., Fritzson, R., McKay, D., & McEntire, R. (1994, November). KQML as an agent communication language. In *Proceedings of the third international conference on Information and knowledge management* (pp. 456-463). ACM.

Forbus, K. D., Ferguson, R. W., Lovett, A., & Gentner, D. (2017). Extending SME to handle large- scale cognitive modeling. Cognitive Science, 41(5), 1152-1201.

Forbus, K. D., Gentner, D., & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive science*, 19(2), 141-205.

Forbus, K. D., & Hinrichs, T. (2017). Analogy and Qualitative Representations in the Companion Cognitive Architecture. *AI Magazine*, 38(4), 34-42.

Ge, Ruifang. *Learning for semantic parsing using statistical syntactic parsing techniques*. Texas Univ. at Austin, 2010.

Hasegawa, D., Cassell, J., & Araki, K. (2010, November). The role of embodiment and perspective in direction-giving systems. In *2010 AAAI Fall Symposium Series*.

He, W., Liu, K., Liu, J., Lyu, Y., Zhao, S., Xiao, X., ... & Liu, X. (2017). Dureader: a chinese machine reading comprehension dataset from real-world applications. *arXiv preprint arXiv:1711.05073*.

Kwiatkowski, T., Zettlemoyer, L., Goldwater, S., & Steedman, M. (2010, October). Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 1223-1233).

Lee, M. K., Kiesler, S., & Forlizzi, J. (2010). Receptionist or Information Kiosk: How Do People Talk with a Robot? *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work* (pp. 31–40).

Liang, P., Jordan, M. I., & Klein, D. (2013). Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2), 389-446.

McCauley, L., & D'Mello, S. (2006, August). MIKI: a speech enabled intelligent kiosk. In *International Workshop on Intelligent Virtual Agents*. Springer, Berlin, Heidelberg.

McFate, C. J., & Forbus, K. D. (2011, June). NULEX: an open-license broad coverage lexicon. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2* (pp. 363-367).

Stocky, T., & Cassell, J. (2002, January). Shared reality: Spatial intelligence in intuitive user interfaces. In *Proceedings of the 7th International Conference on Intelligent User Interfaces* (pp. 224-225). ACM.

Tafjord, O., Clark, P., Gardner, M., Yih, W. T., & Sabharwal, A. (2018). QuaRel: A Dataset and Models for Answering Questions about Qualitative Relationships. *arXiv preprint arXiv:1811.08048*.

Tomai, E., & Forbus, K. D. (2009, March). EA NLU: Practical language understanding for cognitive modeling. In *Twenty-Second International FLAIRS Conference*.

Zelle, J. M., & Mooney, R. J. (1996, August). Learning to parse database queries using inductive logic programming. In *Proceedings of the National Conference on Artificial Intelligence* (pp. 1050-1055).