



Computer Science and Artificial Intelligence Laboratory  
Technical Report

MIT-CSAIL-TR-2018-013

March 1, 2018

---

**A Natural Language Interface for Mobile Devices**  
Boris Katz, Gary Borchardt, Sue Felshin, and  
Federico Mora

# A Natural Language Interface for Mobile Devices

Boris Katz, Gary Borchardt, Sue Felshin, and  
Federico Mora

## Introduction

Humans express their needs almost effortlessly in natural language, and for this reason, constructing machines that can reliably respond to natural language requests has been a longstanding and significant goal in the design of intelligent systems. Early successes—for example, the LUNAR system of Woods, Kaplan, and Nash-Webber (1972); the LADDER system of Hendrix, Sacerdoti, Sagalowicz, and Slocum (1978); and the PLANES system of Waltz (1978)—relied on the use of application-specific grammars to encode various constraints of particular domains or datasets. While this approach enabled these systems to respond to a range of requests in their targeted domains, the systems could not easily be ported to work in related or expanded domains, and they had limited coverage of vocabulary and syntactic constructions used in expressing requests. More recently developed systems—such as those described in Harabagiu, Maiorano, and Pasca (2003); Katz (1997); Nyberg, Burger, Mardis, and Ferrucci (2004); and Weischedel, Xu, and Licuanan (2004)—address multiple domains and employ general-purpose grammars or statistical language interpretation to handle the variety of requests and phrasings of requests that might be submitted by users. With more and more information sources available in networked contexts, plus mobile devices acting as sources and targets of requests, recent systems have adopted a more distributed architecture.

In this chapter, we first discuss some of the primary issues related to the design and construction of natural language interfaces, and in particular, interfaces to mobile devices. Then, we describe two systems in this space: the START information access system and the StartMobile natural language interface to mobile devices. Finally, we discuss recently deployed commercial systems and future directions.

## Goals of Natural Language Interfaces

The primary goal of a natural language interface is, of course, to produce appropriate responses to user requests. The requests may ask the interface for information, or they may ask for actions

to be performed. The interface may issue any number of responses to a single request: it may return several segments of information, for example, or perform several actions.

In general, the appropriateness of an interface's responses can be assessed using variants of the metrics of *recall* and *precision* from the field of information retrieval. In this context, recall can be calculated as the average, over many requests, of the fraction formed by dividing the number of appropriate responses returned by the interface by the total number of appropriate responses possible. Precision can be calculated as the average, over many requests, of the fraction formed by dividing the number of appropriate responses returned by the interface by the total number of responses returned by the interface. Recall is often difficult to compute, as there may be no way to enumerate the full set of possible, appropriate responses to a request—the full set of information segments that might be returned, for example, or all appropriate actions that might be performed in response to a request. Recall can be important but in many cases it is of lesser importance, as in those contexts where the user only needs one or a few satisfying responses. Precision, on the other hand, is almost always important. Information requests that return many inappropriate responses distract the user, waste display space, and may even mislead the user. Inappropriate actions can have serious consequences and must be avoided at all costs.

Another goal of natural language interfaces is domain coverage. Limited-domain systems can be useful in some cases, such as an interface to a single, specific database. In general, however, the recent trend is toward construction of interfaces that provide a single point of interaction with respect to a large quantity of datasets, domains and possible actions.

Ease of use is another important aspect of natural language interfaces. An interface should accept unrestricted natural language input to whatever extent is possible, it should exhibit interactivity for clarification of requests and other purposes, it should respond with multimedia information, it should provide explanations of its behavior when demanded, it should provide a history mechanism for review of past responses, and so forth.

One very important aspect of natural language interfaces is their ability to handle complex requests. The simplest sorts of information requests solicit all available information on a specified topic: e.g., “Tell me about Germany,” or some specific property, e.g., “Tell me about Germany's population distribution.” More complex requests involve the retrieval of information about relationships between entities, or the execution of commands that involve multiple entities. Even more complex are requests that require the system to perform novel analyses or actions. Finally, some requests may involve a nesting of subrequests to be addressed in combination.

Natural language interfaces to mobile devices must exhibit additional characteristics. These interfaces must be able to respond on the basis of information or actions available not only on the mobile device itself but also on systems linked to the device through network connections. Especially where actions are to be performed, such systems must have extremely high precision, as there are often limited ways to rescind these actions. Coupled with this, mobile devices are often used in environments where the user has divided attention and where excessive system interactivity is not desired; thus the interface must in some cases proactively infer what the user has intended, again with extremely high precision. To accomplish this, a system might need to rely on information gleaned from previous interactions with that individual, elements of the preceding dialog, or other available modalities such as camera input. Interfaces to mobile devices must also be particularly responsive to time and location of the user, and these interfaces must make the best use of limited display space.

## START

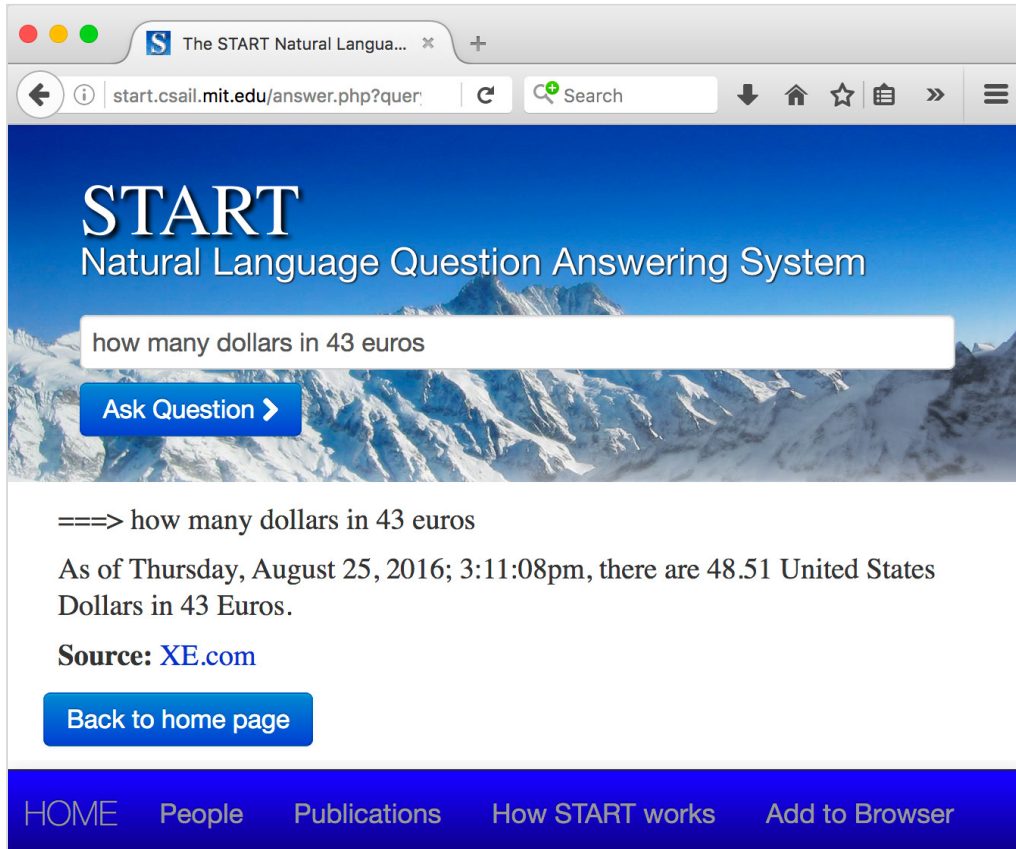
We first discuss the general-purpose START information access system, which has been in development at MIT since the late 1970s (Katz, 1980, 1990, 1997; Katz, Borchardt, & Felshin, 2006). In its most general question-answering application, START is available as a public server at <http://start.mit.edu/>. This version of START answers questions in a range of domains including geography, arts and entertainment, history, science, and the very large number of topics and domains covered in Wikipedia. In addition to the general-purpose public START server, several special-purpose servers have been created for specific topic areas, some of which involve the execution of actions in response to user requests and others of which make use of an API to START's language parsing and generation capabilities. Separately, several strategies pioneered by the START system were incorporated into IBM's Watson, which in 2011 defeated the all-time human champions on the quiz show *Jeopardy!* (Hardesty, 2011; Murdock, 2012).

In its traditional question-answering role, START accepts English questions and offers responses that draw on information sources that include structured, semistructured, and unstructured materials. Some of these materials are maintained locally and some are accessed remotely through the Internet. In some cases, responses are calculated dynamically by the system or its allied resources. A particular emphasis of START is that of providing high-precision information access, such that the user may maintain a high degree of confidence that a response, if returned by the system, is appropriate to the submitted question. Figure 1 presents a sample request–response interaction with START.

A particularly important aspect of START's design is the use of *ternary expressions* as an internal representation of natural language expressions. Ternary expressions represent language as a set of nested subject–relation–object triples, where the subject and object may themselves be ternary expressions (Katz, 1988; Katz, 1990). The ternary expression representation is a versatile syntax-driven representation of language that highlights significant semantic relations and allows for detailed encoding of syntactic and lexical features. It has proved to be extremely beneficial for START's parsing and question answering capabilities due to its speed, compactness, and accuracy for storing, matching, and retrieving information.

As originally configured during the initial stages of its development, START served to answer English questions on the basis of English statements that had been previously submitted to the system, and this operation underlies much of START's current capabilities as well. When START is presented with an English statement for processing, it parses the statement and encodes it in the form of a set of nested ternary expressions. One can think of the resulting entry in START's knowledge base as a “digested summary” of the syntactic structure of the English sentence. User-submitted questions are then analyzed in the same manner and matched against stored assertions in the knowledge base. Matched assertions are then retrieved and expressed as English responses. (The technique of using natural language annotations, described below, extends this approach and enables START to present additional material and perform computations in response to matches.) Because matching occurs at the level of syntactic structures, linguistically sophisticated machinery such as synonymy, hyponymy, ontologies, and structural transformation rules can all be brought to bear on the matching process, thus achieving capabilities far beyond simple keyword matching.

In particular, structural transformation rules enable the system to find matches despite significant differences in expression that arise from alternate realizations of the arguments of



**Figure 1** START performing a currency conversion.

verbs and other constituents (Katz & Levin, 1988). For example, suppose START is presented with a statement

Greece surprised the European Union with its actions.

This statement can also be paraphrased as “*Greece’s actions* surprised the European Union.” In order to match questions related to this alternate version of the statement, START must make use of a structural transformation rule that can be expressed as follows:

```
If      <<subject verb object1> with object2>
Then   <object2 verb object1> AND
       <object2 related-to subject>
Where  verb belongs to the emotional-reaction class
```

With the addition of this rule, START can answer not only questions like

```
Did Greece surprise the European Union with its actions?
Did Greece surprise the European Union?
```

but also, it can answer questions like

```
Did Greece's actions surprise the European Union?  
Which country's actions surprised the European Union?
```

Note that, within START, structural transformation rules are typically associated with classes of verbs, rather than individual verbs. The above rule applies for all verbs in the *emotional-reaction* class, which includes “surprise,” “anger,” “embarrass,” and others. A range of other verb classes suited to use in structural transformation rules may be found in Levin (1993).

A second, significant aspect of START’s design is the use of *natural language annotations* (Katz, 1997). Natural language annotations are natural language phrases and sentences associated with segments of information, describing their content. When START matches a user request to a natural language annotation, it can then access the associated segment of information as a response to the user.

For example, an HTML fragment containing information about clouds on Mars may be annotated with the following English sentences and phrases:

```
clouds on Mars  
Martian clouds are composed of water and carbon dioxide.  
...
```

START parses these annotations and stores the parsed structures (nested ternary expressions) with pointers back to the original information segment. To answer a question, the user query is compared against the annotations stored in the knowledge base. If a match is found between ternary expressions derived from annotations and those derived from the query, the corresponding annotated segment is returned to the user as the answer. For example, annotations like those above allow START to answer the following questions:

```
Are there clouds on Mars?  
Do clouds exist on Mars?  
What is the composition of Martian clouds?  
Do you know what clouds on Mars are made of?  
...
```

Figure 2 presents an example of START answering such a question.

Except for small amounts of particularly vital information, of course, it is impractical to annotate each item of content manually. However, sources of all types—structured, semistructured and unstructured—can contain significant amounts of parallel material. *Parameterized annotations* address this situation by combining fixed language elements with “parameters” that specify variable portions of the annotation. As such, they can be used to describe whole classes of content while preserving the indexing power of nonparameterized annotations. As an example, the parameterized annotation (with parameters in italics)

```
number people live in city.
```

The START Natural Language Question Answering System

What are Martian clouds made of?

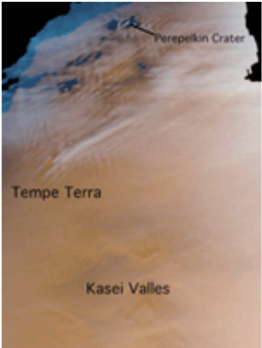
Ask Question >

====> What are Martian clouds made of?

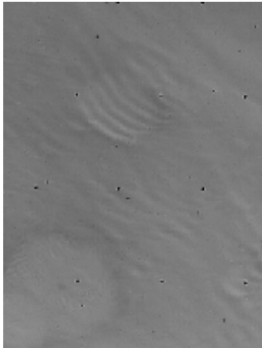
**Clouds**

(Click on any image for a larger view)


Mars has clouds - some are water, some are CO<sub>2</sub>!



MOC image of various types of clouds. (*Malin Space Science Systems/NASA*)



*Viking 2* Orbiter image of wave clouds near the south pole of Mars. This complex pattern of clouds is caused by winds passing over the craters. (*NASA*)



*Viking 1* Orbiter near Mars' northern polar cap showing a cyclonic weather system. (*NASA*)

**Figure 2** START answering a question using annotation-based matching.

can describe, on the data side, a large semistructured Web resource containing population figures for various cities. On the question side, this annotation, supported by structural transformation rules, can recognize questions submitted in many forms:

How many people reside in Chicago?  
 Do many people live in Pittsburgh?  
 What number of people live in Seattle?  
 Are there many people living in Boston?

Additional parameterized annotations may be included that describe the population figures in other ways (for example, using the terms “population” or “populous”), and additional elements of the annotations may be parameterized. As a result, a large number of different questions can be answered using a small number of parameterized annotations. For example, with further parameterization, a single annotation can answer questions about area, elevation, population density, and other properties in addition to population.

The use of natural language annotations, and in particular, parameterized natural language annotations, enables START to respond to user requests in a wide variety of ways. For example, START can retrieve multimedia information or information from resources on the Internet, execute computations, retrieve foreign-language material, and perform specific actions on behalf of its user.

An important subset of retrievable information on the Internet and in structured datasets consists of data that may be viewed as collections of “objects,” with each object having one or more “properties” that have particular “values.” START operates in conjunction with a system called Omnibase that manages information that conforms to this *object–property–value* data model (Katz *et al.*, 2002).

Parameterized annotations serve as the interface between START and Omnibase’s object–property–value data model, allowing the combined systems to answer questions about a variety of topics such as a country’s population, area, GDP, or flag; a city’s population, location, or subway map; or a famous individual’s place of birth, date of birth, or spouse. The object–property–value data model is more generally applicable than it may appear on the surface, as many object–property questions can be cast with diverse phrasing; e.g., “What is Angela Merkel’s date of birth?” can be phrased as “When was Angela Merkel born?” “What is Argentina’s size?” can be phrased as “How big is Argentina?” and so forth. However, there are other possible types of queries that do not fall into the object–property–value model, such as questions about quantities of information that are a function of two objects (e.g., “How can I get from Boston to New York?”). Such questions and the information they request can be modeled through more general natural language annotations.

Figure 3 illustrates a question answered by START, utilizing support from Omnibase.

In order to match input questions to parameterized annotations successfully, START must know which terms can be associated with any given parameter. Omnibase supports this requirement by acting as an external gazetteer for source-specific terminology, with variants of terms being calculated from objects’ names, extracted from semistructured material in information sources, or manually defined. This maintains the integrity of the abstraction layer: information source terminology is kept together with information source processing. Omnibase’s use of the object–property–value data model applies equally well to fixed, semistructured websites and to “deep Web” sources that are accessed through special query languages or interactive form-based interfaces. When START transmits an object–property query to Omnibase, Omnibase executes an access script associated with the information source in question, and the access script may obtain individual elements of information directly from a static Web page, extract data from a local data source, or obtain dynamic information or otherwise “hidden” information by interacting with a query interface.

More recent work has made it possible for the START and Omnibase systems to access information without the need for manually created annotations. Many information sources have a largely regular structure that enables us to extract property–value pairs. In addition, these property names are often in the form of English nouns and other phrases. In such cases, when an



The START Natural Language Question Answering System interface is shown in a browser window. The page title is "START Natural Language Question Answering System". A search bar contains the question "Show me a map of the Chicago subway system." Below the search bar is a blue button labeled "Ask Question >".

The system's response is displayed below the search bar:

====> Show me a map of the Chicago subway system.

*Chicago, Illinois, USA*

Here is a map of the metro system of Chicago, Illinois, USA:



2012 © UrbanRail.Net (R. Schwandl)  
Click on map to expand!

Source: [UrbanRail.net](http://UrbanRail.net)

**Figure 3** START and Omnibase answering a question using material from an English source.

object–property–value question is asked, START first analyzes the question to find the object and property names, and then runs a procedure to extract the value as a response to the question. Using this technique, START can automatically answer questions such as “What was Einstein’s alma mater?” or “What is the calling code of Italy?”

One particularly useful source of information is Wikipedia, the world’s largest crowdsourced encyclopedia with over five million articles. Articles are organized in hierarchical sections, and many have an “infobox,” a table that summarizes key information in the article. To access these kinds of information, we developed WikipediaBase (Morales, 2016), a system that turns Wikipedia into a virtual database and organizes it in an object–property–value data model. We consider infobox attributes and section headers to be properties. Using WikipediaBase, START is able to respond to requests like “Tell me about Albert Einstein’s personal life” with the contents of the “Personal life” section, or “What awards did Einstein receive?” with the “Notable awards” row in Albert Einstein’s infobox.

With its understanding of English morphology and syntax, START can recognize variations of object–property–value questions. It can correctly answer questions such as “Who designed the Oakland Bay Bridge?” (from the “Designer” property in the infobox) or “What river does the Brooklyn Bridge cross?” (from “Crosses: East River” in the infobox).

It is also possible to ask about information in ways that share little surface similarity with the property names. For instance, “Which college did Albert Einstein attend?” and “Where did Einstein study?” are valid paraphrases of “What is the alma mater of Albert Einstein?” but they share few content words. To address these types of questions, we compiled a crowdsourced corpus of over 15,000 questions about Wikipedia infoboxes. We used these questions to train a machine learning model that selects the correct response from a set of candidate answers with high accuracy (Morales, 2016; Morales, Prentoon, Avery, Felshin, & Katz, 2016). Our ongoing work in automatic techniques to answer questions will allow the START system to quickly scale up to new types of questions and information sources.

Some user requests may contain subrequests. For example, a user of START might submit a request “When was the president of France born?” Such questions are interesting because answering them typically involves information from different sources, and indeed, a system must answer one part of the question—e.g., “Who is the president of France?”—before proceeding to use the answer to that subquestion—in this case, François Hollande—within another subquestion to be answered—e.g., “When was François Hollande born?” Natural language annotations can help, in that they can be used to describe sets of simple questions that can be answered independently. In addition, the mechanism of parameter matching—via synonyms, hyponyms, etc.—plus the underlying mechanisms that supply answers to the simple questions, can be used to bridge terminology differences between resources, permitting a range of complex questions to be answered.

START utilizes an approach in which it analyzes complex questions linguistically in order to isolate valid candidate subquestions and determine an appropriate order in which to answer those subquestions. START then checks, via its base of annotated resource materials, to see if particular subquestions can be answered. This approach is described more fully in Katz, Borchardt, and Felshin (2005). Figure 4 provides an example of START answering a complex question.

START also contains a sophisticated natural-language-generation capability, which takes a set of ternary expressions as input and converts this set into readable English. In addition, prior to generation, ternary expressions can be joined together, modified, and augmented by the system, enabling START to produce individual sentences, narrative text, and dialog elements as appropriate.

Taken together, START’s collection of representations and techniques—its ternary expressions, structural transformation rules, natural language annotations, syntactic decomposition

The START Natural Language Question Answering System

# START

## Natural Language Question Answering System

What are the military expenditures of the 2 most populous countries in Europe?


[Ask Question >](#)

====> What are the military expenditures of the 2 most populous countries in Europe?

*I know that two most populous populations in Europe are Russia and Germany (source: [The World Factbook](#)).*

*Using this information, I determined:*

### Russia




Military expenditures:  
 3.49% of GDP (2014)  
 3.18% of GDP (2013)  
 2.92% of GDP (2012)  
 2.71% of GDP (2011)

**Source:** [The World Factbook](#)

*Using this information, I determined:*

### Germany



Military expenditures:  
 1.18% of GDP (2015)  
 1.35% of GDP (2012)  
 1.34% of GDP (2011)  
 1.35% of GDP (2010)

**Source:** [The World Factbook](#)

**Figure 4** START answering a complex question using its syntactic decomposition strategy.

strategy, natural-language-generation capability, and so forth—provide a platform for interpreting a range of requests and issuing a range of responses to requests. Natural language annotations, in particular, enable START to execute arbitrary procedures in response to requests, and this allows START to serve not only as a question answering system but also as an interface through which user requests can result in virtual or physical actions. The StartMobile system, described next, is one such application in which START is used to perform actions on a mobile device on behalf of its user.

## StartMobile

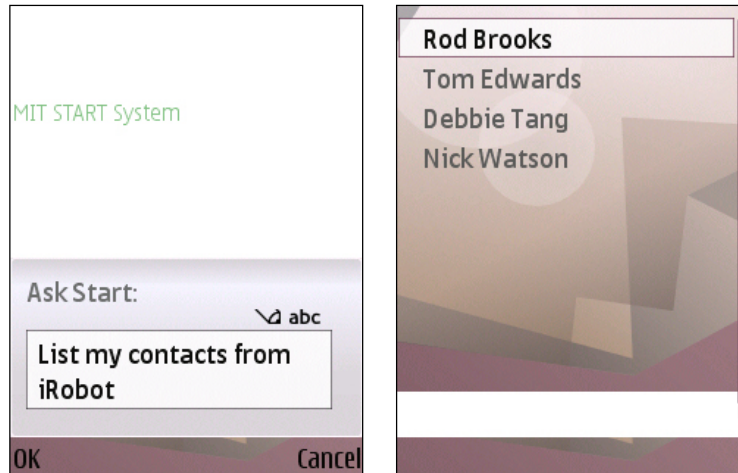
In an application that foreshadowed the introduction of systems such as Apple’s Siri, we used START to create a system called StartMobile, which provides a natural language interface to mobile devices (Bourzac, 2006; Katz, Borchardt, Felshin, & Mora, 2007; Katz, Mora, Borchardt, & Felshin, 2011). StartMobile allows its users to pose English requests for information present on their mobile devices, issue commands to perform actions on their devices, and make requests for information available from a broad range of sources beyond the confines of their device. Requests may be entered in written form, or by voice, using speech recognition utilities offered by Google, Inc.

StartMobile uses the START system as a first stage in the processing of user requests. START performs an initial interpretation of the requests, and if these requests concern the retrieval of general information from the World Wide Web or other sources, START obtains the information for presentation to the user. If it is not possible to complete the interpretation of the requests, however, or if the requests involve actions that must be performed on the user’s mobile device, START encodes the user’s requests in a language called Moebius, which has been designed to convey natural language requests in various stages of interpretation between systems and devices. Finally, software that resides on the user’s mobile device completes the interpretation of user requests, if necessary, and performs required actions to fulfill those requests.

The StartMobile system supports a range of activities on several models of mobile phones:

- retrieving general-purpose information for the benefit of the user;
- retrieving contact and calendar information stored on the user’s mobile device;
- retrieving text messages and managing the user’s text message inbox;
- placing phone calls from the user’s mobile device;
- creating reminders on the user’s mobile device or on other users’ mobile devices;
- taking pictures with the mobile device’s camera;
- modifying device settings;
- accessing position information and displaying associated map and direction information on the user’s mobile device; and
- retrieving video tutorials for presentation to the mobile user.

Figure 5 illustrates the StartMobile system in action, using typewritten entry of requests. (To place StartMobile in its appropriate historical context as a precursor to today’s commercial systems, this and following screenshots show phone output captured in 2006–2007 during



**Figure 5** StartMobile performing a search within the contacts stored on a mobile device.

StartMobile’s initial development.) In the interaction depicted in Figure 5, the mobile user has asked the system to list the user’s contacts at a specific company.

Through the use of parameterized annotations, structural transformation rules, and related technology, START enables the StartMobile application to accept requests in a range of variant forms. For the example illustrated in Figure 5, some of these variant forms are:

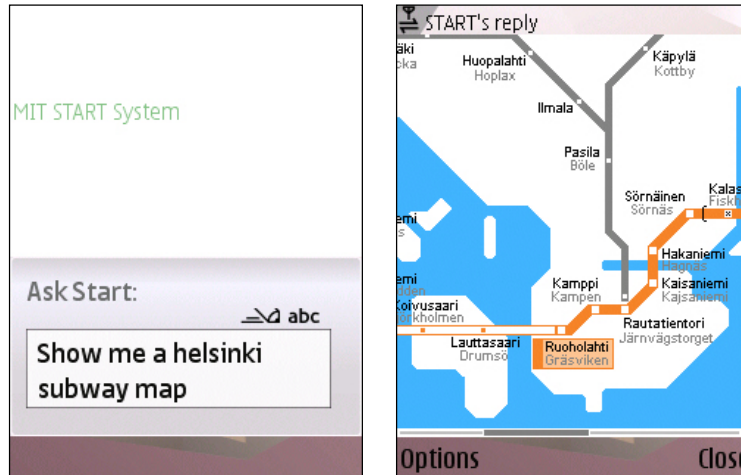
Who works for iRobot?  
 Who do I know at iRobot?  
 Which of my friends work at iRobot?  
 Show my colleagues from iRobot.

START is used for general information access in the StartMobile system. START’s answers to general questions are streamlined for display on small screens, then relayed to the user’s mobile device for presentation to the user. Figure 6 illustrates StartMobile used to retrieve multimedia information. An alternative mechanism within StartMobile allows users to submit text messages to the START server, then also receive responses by text message.

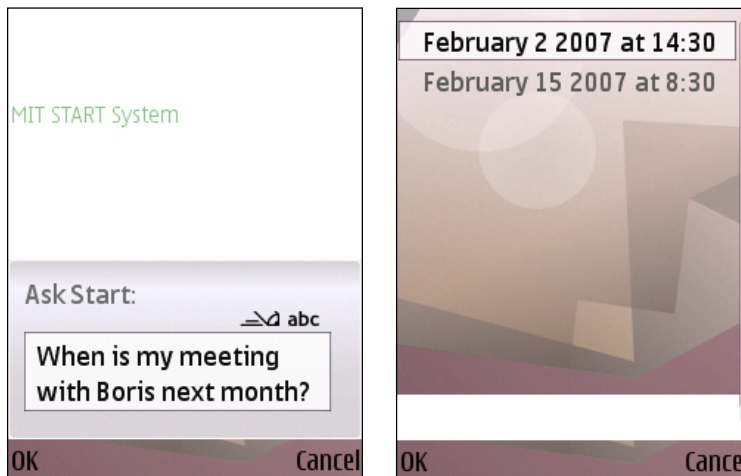
Figures 5 and 6 depict fully grammatical requests submitted by the user. StartMobile is also configured to allow the user to enter fragmentary utterances in a range of cases where the meanings remain clear. For the request illustrated in Figure 6, for example, the user could have entered “Helsinki subway map” or “subway map Helsinki,” resulting in a display of the same map.

Other types of requests concern information maintained on the user’s mobile device. To handle these requests, START matches them to natural language annotations as always; however, the annotated material in this instance is a procedure that relays instructions to the mobile device. Associated software on the mobile device performs the necessary operations and delivers the results to the user. Figure 7 depicts the handling of such a request, involving a search through the calendar on the user’s mobile device.

Some user requests may contain unusual names—people, streets, cities, businesses, etc.—that appear within particular data entries on the user’s mobile device. To enable START to correctly



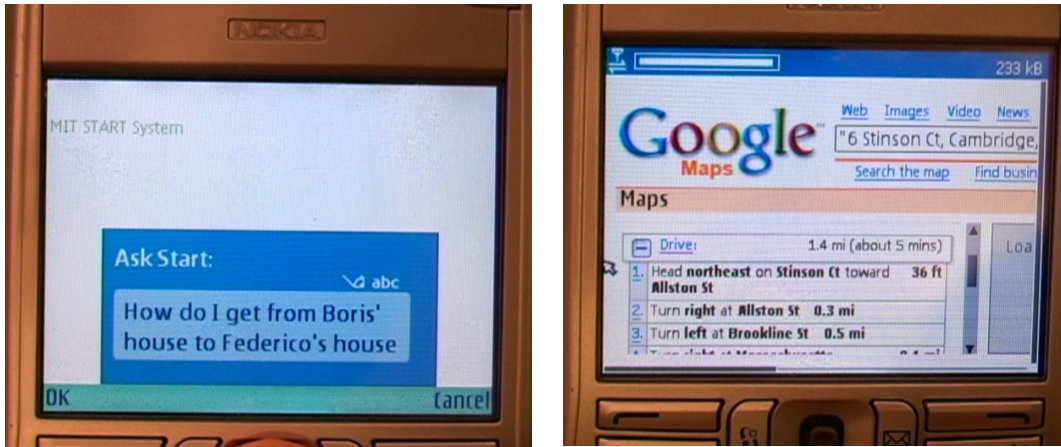
**Figure 6** StartMobile responding to a request for general information.



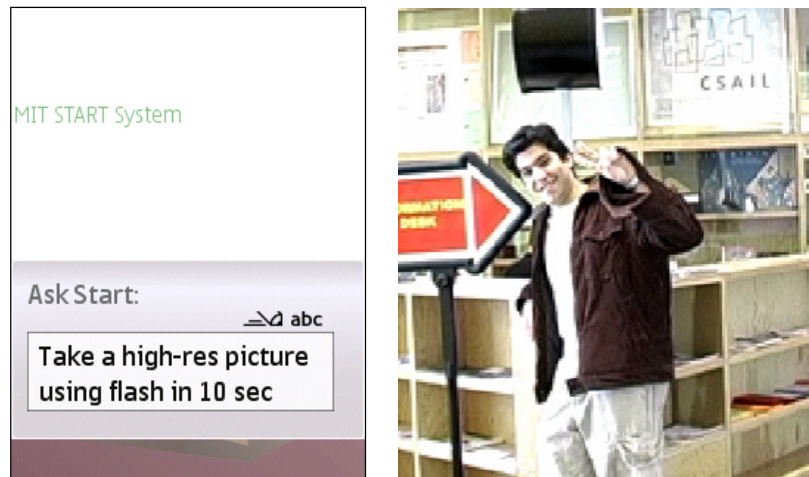
**Figure 7** StartMobile searching through the calendar on the user's mobile device.

analyze these requests and take appropriate actions, StartMobile implements a mechanism whereby submitted user requests are initially inspected, on the user's mobile device, to recognize and categorize names that appear in data sets such as the contacts database or calendar. START uses this information in a manner parallel to its use of the Omnibase system as a gazetteer. For the request illustrated in Figure 8, this mechanism enables StartMobile to correctly process the names "Boris" and "Federico."

In another set of cases, the user's input is not a request for information but rather a command to perform an action on the user's mobile device. These requests are handled in a similar manner to requests for information on the user's mobile device, with START relaying instructions to software that performs actions on the user's mobile device. Figure 9 illustrates StartMobile's handling of a request to take a picture using the camera on the user's mobile device.



**Figure 8** StartMobile presenting directions from one location to another location.

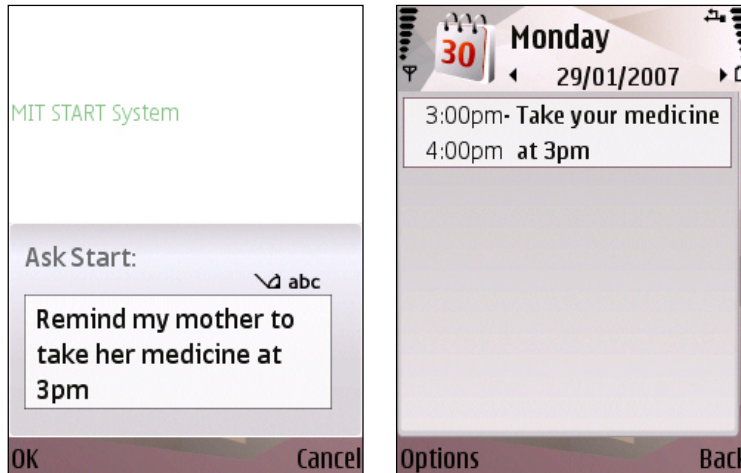


**Figure 9** StartMobile responding to a request to take a picture using a phone's camera.

In still other cases, the user may enter a request on one mobile device to perform an action on another mobile device. In this instance, START will relay instructions to the first device, which must then relay appropriate instructions to the affiliated device. Figure 10 presents an example of this type of request being handled by StartMobile.

Within the StartMobile application, high precision is extremely important. The system is frequently asked to perform actions that may not easily be rescinded. Separately, in some circumstances interactivity may be less desirable than for, say, a user interacting through a computer console. Finally, limited display space increases the inconvenience caused by inappropriate responses in listed results.

Another significant issue for natural language interfaces to mobile devices arises from the distributed nature of processing in this context. It is often the case that natural language requests can only be fully understood—their ambiguities resolved—in the presence of specific, matching



**Figure 10** StartMobile posting a reminder on an affiliated mobile device.

components of knowledge. In distributed environments, this knowledge is distributed, requiring the networked devices and systems in some cases to collaborate not only toward the ultimate satisfaction of the received requests but also toward the initial understanding of requests, so that it is possible to satisfy them.

StartMobile makes use of an intermediate, language-based representation called Moebius (Borchardt, 2014), which supports distributed interpretation and distributed fulfillment of natural language requests. Moebius serves to encode natural language requests at varied stages of syntactic and semantic interpretation, so that these requests may be relayed between systems—for instance, a user’s mobile device, central servers, and other users’ mobile devices—in order to receive additional interpretation and fulfillment. While Moebius specifically addresses the representation and processing of ambiguous requests, it is also applicable for more straightforward requests and thus we use the language as an intermediate representation for all StartMobile requests that must be relayed between systems and devices.

Following is an example of a Moebius expression issued by START to the user’s mobile device, depicting a substantially interpreted version of the English request illustrated in Figure 10.

```

alert(object:person mother(of:person "user"),
      with:message_string
        "Take your medicine at 3pm.",
      at:time "2007-01-29T15:00:00")!

```

Moebius specifies basic syntactic relations between elements of the representation, and it adds semantic labels, drawn from a hierarchy of general to specific categories.

A key aspect of Moebius is that it uses language itself as a representation. In this respect, it shares a common orientation with the START system. START uses language-based ternary expressions as a representation for both questions and natural language annotations. Indeed, when START matches a question to a natural language annotation it does two things: it provides an answer to the question, and it commits to an interpretation of the question. Moebius can be



thought of as extending this idea to distributed contexts, enabling partially interpreted requests to be interpreted and fulfilled by collective action on the part of multiple systems in a distributed environment.

As an example of the use of Moebius to characterize an ambiguous request at different stages of interpretation, consider the request

```
Is Carl at IBM?
```

This question could be offered to ascertain whether or not Carl is employed by IBM, or it could be offered to determine whether or not Carl is, at the moment, physically present at an IBM facility. We assume that the human user has constructed the request in compliance with conversational maxims such as proposed by Grice (1975)—that is, by supplying an adequate amount of information, but not too much information, by being truthful, by supplying only relevant information, and by being clear or perspicuous. To the human user, the request “Is Carl at IBM?” may be unambiguous in context; however, the system may need additional knowledge to disambiguate the request. The system may obtain this knowledge by consulting the repertoire of capabilities offered by components expected to fulfill the request (that is, whether these components are known to be able to respond to one interpretation or the other), by referencing contextual information from the current state of processing, by consultation with the human user, and so forth.

If the device that initially processes the request “Is Carl at IBM?” does not have access to the knowledge needed to fully interpret the request, then, using Moebius, that device can encode the request in a partially interpreted form:

```
be(subject:person "Carl", at:object "IBM")?
```

This representation parses the request syntactically, yet it makes no commitment as to the semantic interpretation of the relationship between “Carl” and “IBM” or as to the specific semantic category of “IBM” (“object” being the most general semantic category). If this request is relayed to another device or system that possesses the necessary knowledge to disambiguate the request, that system may cast the request into one of two more fully interpreted forms. If the determination is made that the request concerns physical presence at an IBM facility, the request can be reexpressed as

```
be(subject:person "Carl", at:facility "IBM")?
```

where “IBM” is classified semantically as a physical “facility.” On the other hand, if the determination is made that the request concerns employment, the request can be reexpressed as

```
employ(subject:organization "IBM",  
        object:person "Carl")?
```

where “IBM” is classified semantically as an abstract “organization,” and the relationship is reexpressed as one of employment. Subsequent processing can then continue according to the chosen interpretation.

In StartMobile, when the mobile device receives the partially interpreted form of the request “Is Carl at IBM?” it chooses to interpret this as a request about employment. As a result, it translates the received expression into the more fully interpreted Moebius expression requesting

employment information for Carl, then processes this Moebius expression. Figure 11 illustrates StartMobile’s handling of the request “Is Carl at IBM?” using employment information recorded in the contacts database of the user’s mobile device. The displayed output in this case explicitly informs the user that StartMobile has retrieved employment information, so as to clarify StartMobile’s interpretation of the user’s request.

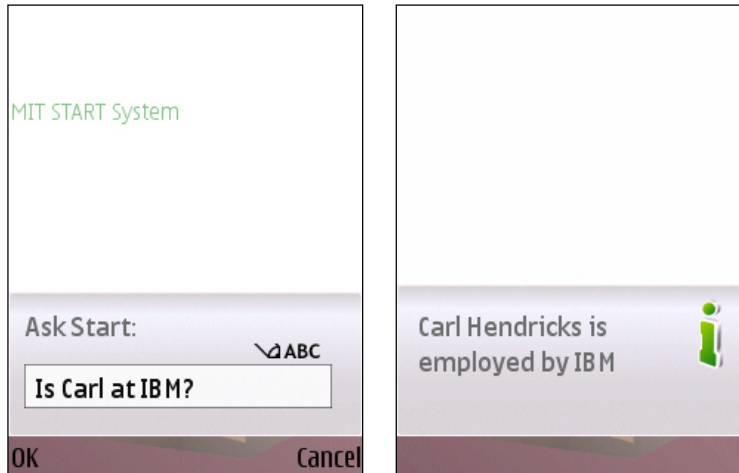
In general, ambiguity can arise from many sources—abstract verbs; syntactic ambiguities; omitted adverbial phrases; ambiguous prepositions and conjunctions; abstract semantic categories; descriptions of objects; ambiguous names, times, and places; anaphora; and abstract adjectives and adverbs, to name a few. Moebius provides several mechanisms for depicting and resolving these ambiguities. Abstract verbs can be replaced by more specific verbs. For example, a request to “contact” a person can be reexpressed as a request to “call” a number or “send” a message. Abstract semantic categories can be replaced by more specific categories. For example, “message” can be replaced by “e-mail\_message” “text\_message” “voice\_message” and so forth. Descriptive subexpressions such as “address(of:facility apartment(of:person "Sandra"))” and “3 o'clock” can be replaced by more specific expressions such as “298 Beacon Street, Boston, MA 02116” and “2013-07-22T15:00:00” In addition, ambiguous commands, statements, and questions can be clarified by inserting adverbial phrases, for example, or replacing the original expressions with entirely different expressions. The goal of Moebius is to capture a range of such ambiguities in various stages of interpretation, and we have found that simple natural language, structured for ease of computer processing, provides sufficient expressiveness to model many common ambiguities.

The StartMobile system situates START as a central server, accessed by one or more mobile devices that occasionally interact directly with one another. START performs initial processing of user requests, then passes Moebius requests to other systems and devices as needed for further interpretation and/or fulfillment. However, the overall design of the StartMobile system allows for other configurations as well.

One alternative is to perform initial processing of natural language requests on each user’s mobile device. Each mobile device could then relay Moebius requests to other devices and systems as necessary for further interpretation based on knowledge held by those systems, or for the completion of actions external to that mobile device.

A third possibility involves a mixture of these two approaches. In this configuration, each user’s mobile device would contain a lightweight capability for simple language processing, then pass off partially interpreted requests or even uninterpreted requests to other, more substantial language processing components as the need arises.

A particular emphasis of StartMobile has been enabling users to pose requests that result in actions: setting reminders, for example, or taking pictures or setting location-triggered alerts. Subsequent to the StartMobile effort, we have continued our exploration into the construction of natural language interfaces that carry out actions on behalf of their users. In particular, we used START as one component in a system called the Analyst’s Assistant, which supports collaborative user–system interpretation of vehicle events exhibited in a dataset of vehicle track information (Borchardt *et al.*, 2014). From this and related efforts, we have arrived at the view that natural language interfaces that perform actions on behalf of their users can benefit greatly from targeted support for performing *collections* of interrelated actions. In the mobile phone domain, for example, a user might wish to use the interface to take a picture, then send it to a friend, then attach a label to the picture and save it in the phone’s memory. Particular capabilities that can enhance the effectiveness of an interface in supporting these kinds of



**Figure 11** StartMobile responding to an ambiguous request.

interactions are: (a) having a robust referencing capability, where the user can refer to previously mentioned quantities or previous responses of the system using simple constructions like “that picture,” “my church friends,” or mouse/touchscreen selections, and (b) providing support, both in input and output, for requests and descriptions at multiple levels of granularity, from coarse-granularity actions and returned summaries to fine-granularity specifications and “drill-down” results.

## Commercial Systems

START and StartMobile are largely rule-based in construction, with these rules created through some amount of human involvement and effort. This enables the systems to respond to rather complex requests, well beyond the range of entity description and relationship requests that can be supported by simple keyword-based or statistical interpretation techniques. This has also enabled these systems to achieve very high precision in their responses. On the other hand, with the explosion of information available on the Web and maintained within mobile devices, it is difficult to provide comprehensive coverage of available sources and request types without some amount of automated construction of capabilities. Current commercial systems such as Apple’s Siri, IBM’s Watson, Google’s “Google Now,” Microsoft’s Cortana, and Amazon’s Alexa employ technology of the sort contained in START and StartMobile in combination with statistical interpretation and calculation of responses based on large-scale machine learning. This provides additional coverage for these systems, plus increased ability to handle ill-formed or idiosyncratic requests, at the expense of some amount of precision in the production of responses.

Current commercial systems may accept either written requests, speech input, or both. Systems that accept both written and speech inputs are typically designed in a modular fashion, as is the case with StartMobile, where speech input is independently processed and the results of speech recognition are submitted to the question answering component of the system.

Capabilities offered by Google, Inc., can serve as an illustration of current design practice in the construction of information access and question answering systems. While these capabilities provide broad coverage by using statistical machine learning techniques to match requests directly to potentially relevant material in unstructured sources, it is also the case that considerable attention has been given to *a priori* structuring of knowledge found in various sources, similar in spirit to the object–property–value structuring of knowledge in START’s companion system Omnibase, leading to higher precision answering of particular types of requests (Dong *et al.*, 2014).

## Conclusion

Cellular telephones and other mobile devices have the potential to provide users with many useful features and capabilities, but the more capable these devices become, the harder it becomes to make use of them with traditional interfaces. Natural language can express a wide range of requests in a very compact form, intuitively usable by humans. Natural language interfaces thus have the potential to significantly reduce the complexity of interaction with mobile devices. We view recent advances in the construction of natural language interfaces as welcome steps towards this goal.

A number of challenges remain in the design and construction of natural language interfaces to mobile devices:

- Further integration is needed between rule-based processing techniques, such as those that produce the high precision of responses in START and StartMobile, with large-scale machine learning technology, such as that which produces the domain coverage of current commercial systems.
- Many systems exhibit limited coverage of requests involving complex syntactic constructions, assumed context, and special-purpose vocabulary.
- Where speech input is accepted, tighter integration of the speech recognition and request processing components may be possible, so that the interpretation of speech input can to a larger extent be influenced by capabilities and constraints of the request processing component.
- Current systems rarely exhibit the ability to explain the manner in which their responses have been determined. Such explanations would help users assess the likelihood that a system’s responses are appropriate to the submitted request.
- Finally, additional work is required in supporting distributed processing of requests, where multiple devices and systems hold pieces of information that are needed for interpretation and fulfillment of the requests.

## Acknowledgements

The work described in this chapter has been supported in part through funding provided by the Defense Advanced Research Projects Agency, Nokia Corporation, and the Intelligence Advanced Research Projects Activity; in part by AFRL contract No. FA8750-15-C-0010; and in part by the Center for Brains, Minds, and Machines (CBMM), funded by NSF STC award CCF-1231216. The authors also wish to thank Alvaro Morales for assistance with this chapter.

## References

- Borchardt, G. C. (2014). *Moebius language reference, version 1.2* (Report MIT-CSAIL-TR-2014-005). Cambridge, MA: MIT Computer Science and Artificial Intelligence Laboratory.
- Borchardt, G., Katz, B., Nguyen, H.-L., Felshin, S., Senne, K., & Wang, A. (2014). *An analyst's assistant for the interpretation of vehicle track data* (Report MIT-CSAIL-TR-2014-022). Cambridge, MA: MIT Computer Science and Artificial Intelligence Laboratory.
- Bourzac, K. (2006, April 27). Nokia phones go to natural language class. *Communications News, MIT Technology Review*. Retrieved from <http://www.technologyreview.com/news/405713/nokia-phones-go-to-natural-language-class/>
- Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K.,... Zhang, W. (2014). Knowledge Vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 601–610). New York, NY: ACM.
- Grice, H. P. (1975). Logic and conversation. In P. Cole & J. L. Morgan (Eds.), *Syntax and Semantics, Volume 3: Speech Acts* (pp. 41–58). New York, NY: Academic Press.
- Harabagiu, S. M., Maiorano, S. J., & Pasca, M. A. (2003). Open-domain textual question answering techniques. *Natural Language Engineering, 9*(3), 1–38.
- Hardesty, L. (2011, April 19). The brains behind Watson. *MIT News Magazine, MIT Technology Review*. Retrieved from <http://www.technologyreview.com/article/423712/the-brains-behind-watson/>
- Hendrix, G., Sacerdoti, E., Sagalowicz, D., & Slocum, J. (1978). Developing a natural language interface to complex data. *ACM Transactions on Database Systems, 3*(2), 105–147.
- Katz, B. (1980). *A three-step procedure for language generation* (A.I. Memo 599). Cambridge, MA: MIT Artificial Intelligence Laboratory.
- Katz, B. (1988). Using English for indexing and retrieving. In *Proceedings of the 1st RIAO Conference on User-Oriented Content-Based Text and Image Handling (RIAO '88)* (pp. 313–333). Paris: Centre de Hautes Etudes Internationales d'Informatique Documentaire (CID).
- Katz, B. (1990). Using English for indexing and retrieving. In P. H. Winston & S. A. Shellard (Eds.), *Artificial intelligence at MIT: Expanding frontiers* (vol. 1, pp. 134–165). Cambridge, MA: MIT Press.
- Katz, B. (1997). Annotating the World Wide Web using natural language. In *Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet (RIAO '97)* (pp. 136–155). Paris: Centre de Hautes Etudes Internationales d'Informatique Documentaire (CID).
- Katz, B., Borchardt, G., & Felshin, S. (2005). Syntactic and semantic decomposition strategies for question answering from multiple resources. In *Proceedings of the AAAI 2005 Workshop on Inference for Textual Question Answering* (pp. 35–41). Menlo Park, CA: AAAI Press.
- Katz, B., Borchardt, G., & Felshin, S. (2006). Natural language annotations for question answering. In *Proceedings of the 19th International FLAIRS Conference (FLAIRS 2006)* (pp. 303–306). Menlo Park, CA: AAAI Press.
- Katz, B., Borchardt, G., Felshin, S., & Mora, F. (2007). Harnessing language in mobile environments. In *Proceedings of the First IEEE International Conference on Semantic Computing (ICSC 2007)* (pp. 421–428). Los Alamitos, CA: IEEE Computer Society Conference Publishing Services.
- Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J., Marton, G.,... Temelkuran, B. (2002). Omnibase: Uniform access to heterogeneous data for question answering. In *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB 2002)* (pp. 230–234). Berlin: Springer.
- Katz, B., & Levin, B. (1988). Exploiting lexical regularities in designing natural language systems. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING '88)* (pp. 316–323). Budapest: John von Neumann Society for Computing Sciences.

- Katz, B., Mora, F., Borchardt, G., & Felshin, S. (2011). *StartMobile: Using language to connect people to mobile devices* [Video File]. Retrieved from <https://www.youtube.com/watch?v=BqOKqXaUWOW>
- Levin, B. (1993). *English verb classes and alternations: A preliminary investigation*. Chicago, IL: University of Chicago Press.
- Morales, A. (2016). *Learning to answer questions from semi-structured knowledge sources*, Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA. Retrieved from <https://dspace.mit.edu/handle/1721.1/105973>
- Morales, A., Premtoon, V., Avery, C., Felshin, S., & Katz, B. (2016). Learning to answer questions from Wikipedia infoboxes. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)* (pp. 1930–1935). Stroudsburg, PA: Association for Computational Linguistics.
- Murdock, J. W. (2012). This is Watson. *IBM Journal of Research and Development*, 56(3–4).
- Nyberg, E., Burger, J., Mardis, S., & Ferrucci, D. (2004). Software architectures for advanced QA. In M. Maybury (Ed.), *New directions in question answering* (pp. 19–29). Cambridge, MA: MIT Press.
- Waltz, D. L. (1978). An English language question answering system for a large relational database. *Communications of the ACM*, 21(7), 526–539.
- Weischedel, R., Xu, J., & Licuanan, A. (2004). A hybrid approach to answering biographical questions. In M. Maybury (Ed.), *New directions in question answering* (pp. 59–69). Cambridge, MA: MIT Press.
- Woods, W. A., Kaplan, R. M., & Nash-Webber, B. L. (1972). *The Lunar Sciences Natural Language Information System: Final report* (Report No. 2378). Cambridge, MA: BBN Technologies.

