Lecture 15
Programming Virtual Assistants in Natural Language

1. Open federated virtual assistants
2. Programming in natural language
3. ThingTalk: Virtual assistant programming language
4. Natural language processing
5. Sharing with privacy
6. Automatic generation of GUIs

Papers in http://oval.cs.stanford.edu

1. Today’s Virtual Assistants

- Amazon Alexa, Google Assistant
- Stores users’ credentials to web accounts & IoTs
- Uniform, personalized natural-language based interfaces
- Repository: 40,000 skills (Amazon), 1 million functions (Google)
- Are we witnessing the creation of the proprietary linguistic web?
- Will virtual assistants become platform monopolies? Open competition?
- Will monopolies own billions of personal information? Privacy?
- AI can predict human behavior & automate work flow: will AI be limited to platform monopolies?
Almond: Open, Federated Virtual Assistants

Runs on your own devices or the web

2. Programming Virtual Assistants in Natural Language

- All our data are silo’ed in different services/devices
- Can we compose operations from different services?
- Can we use assistants to help us control
  - “who, what, when, where, how” to share data?
Example: Asthma Patient

People
Dr. Smith: “if Bob’s peak flow-meter drops below 180L/min let me know”

Environment
Dr. Smith: “when the air quality index is above 500 and Bob is running, warn him”

Location
Bob: “Let my Dad know if I am at the hospital”

Devices
Bob: “when I use my inhaler, record my GPS location in logfile on Box”

Impact of Natural Language Programming

• Natural language: highest-level programming languages
• Gives the power of programming to end users!
• Today’s software cannot satisfy the long tail of user needs
3. ThingTalk: Virtual Assistant Programming Language

“When I use my inhaler, get my GPS location, if it is not home, write it to logfile in Box.”

• Event-driven program
• Multiple function calls
• Parameter passing
• Filters on values
Almond: 1st Programmable Virtual Assistant

Natural Language Commands → LUInet (Linguistic User Interface Neural Network) → Formal target language

```
When I use my inhaler, get my GPS location, if it is not home, write it to logfile in Box.
```

```
monitor @Inhaler-use(), => @GPS(), location => "home"
=> @Box-write(file="logfile", data=location)
```

Thingpedia: Encyclopedia of Things

- Interoperability
  - API signatures + corresponding NL
  - Not just intents
- Open repository
  - Available to Alexa, Google Assistant, …

> 60 devices / 200 functions

<table>
<thead>
<tr>
<th>Twitter</th>
<th>Natural Language</th>
<th>API Signatures</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHEN</td>
<td>@Stanford tweets</td>
<td>Monitor (@home_timeline(), …) author==&quot;Stanford&quot;)</td>
</tr>
<tr>
<td>GET</td>
<td>tweets matching &quot;#Cardinal&quot;</td>
<td>search(...), contains (hashtag, …)</td>
</tr>
<tr>
<td>DO</td>
<td>tweet &quot;Stanford won!&quot;</td>
<td>post (status)</td>
</tr>
</tbody>
</table>
**ThingTalk Compound Statement**

**WHEN** [FILTERS]  \( \rightarrow \) **GET** [FILTERS]  \( \rightarrow \) **DO**

**FILTERS:** =, <, >, <=, >=, <>, contains, starts with, ends with

- When I use my inhaler, get my location, save them to Dropbox
- If I get taken to a hospital, let my dad know.
- When the air quality index is above 500, and I am running, send me an SMS.

- When the Bitcoin price reaches $10,000, search for a “bitcoin” picture, and tweet it with caption “I’m rich!”

---

**Execution of ThingTalk**

Show me tweets with hashtag #Stanford

TweetHashtagFilter(HashTag : String) {
    @twitter.source(text, hashtags, _, from, _, false),
    $contains(hashtags, HashTag)
\( \Rightarrow \) @$notify("Tweet from @" + from + ": " + text);
}

1. Show tweet
2. thingtalk
3. get my "twitter"
4. get twitter interface
5. return twitter interface
6. Oauth
7. return authenticated twitter instance
8. interface with my Twitter
9. report event
10. notify user
Expressiveness of ThingTalk

- Inspired by IFTTT
- ThingTalk is a superset of IFTTT recipes (2 clauses)
- IFTTT has 250,000 unique recipes
- IFTTT provides a GUI: no formal or natural language
- IFTTT is proprietary: user must share credentials

4. Real Natural Language Input

When my car is at home, and it is not plugged in, send me a reminder email

Remind me if my car is not plugged in at home.
If I am not charging my car when it is home, let me know.
Remind me to plug in my car whenever I’m home.
Technical Challenges

• Co-design of Thingpedia, ThingTalk and LUInet
• Natural language training-data acquisition → methodology, tool: Genie

Genie: A Semantic Parser Generation Tool

<table>
<thead>
<tr>
<th></th>
<th>Natural Language</th>
<th>API Signatures</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHEN</td>
<td>@Stanford tweets</td>
<td>Monitor (@home_timeline(), ...)</td>
</tr>
<tr>
<td>GET</td>
<td>tweets matching &quot;#Cardinal&quot;</td>
<td>search(...), contains (hashtag, ...)</td>
</tr>
<tr>
<td>DO</td>
<td>tweet &quot;Stanford won!&quot;</td>
<td>post (status)</td>
</tr>
</tbody>
</table>

Thingpedia

ThingTalk Grammar

NL Templates
  When <> get <>
  Get <> when <>

Synthesize programs

Paraphrase

Parameter & data augmentation

Training Data

LUInet

Code
LUInet Results

- Dataset (44 skills, 122 functions, 147 parameters)
  - 664K synthetic, 24K paraphrased sentences
  - Training set after parameter expansion: 2.9 M sentences
- Model: DecaNLP — MQAN (Multi-Task Question Answering Network):
  - Encoder (bottom to top):
    - BiLSTM
    - Co-attention (context-question)
    - BiLSTM
    - Self-Attention
  - Decoder:
    - LSTM
    - Self-Attention
    - Encoder-Decoder Attention
    - Mixed pointer-generator layer
- Accuracy (current):
  76% on paraphrases; 59% on user input, based on the cheat sheet

5. Sharing with privacy

Let your virtual assistant help you share
**General+Fine-Grain: ThingTalk Extension**

Requester:  \[\text{GET-PREDICATE} \ [\text{FILTERS}]\]

\[\text{WHEN} \ [\text{FILTERS}] \rightarrow \text{GET} \ [\text{FILTERS}] \rightarrow \text{DO}\]

**FILTERS**: $=, <, >, <=, >=$, contains, starts with, ends with

Let Dr. Smith monitor my peak-flow-meter, if it drops below 180L/min

Let my father monitor my security camera for motion,

Let my secretary, whenever I am out of town, read email messages whose subject is marked urgent.

Let my daughter, from 6-8pm, watch Netflix

Let my boyfriend get pictures from my dropbox, taken on Feb 14, and post them on Facebook

---

**Almond: 1st Federated Virtual Assistant**

**Expressiveness:**
Any ThingTalk command

**Privacy:**
Remote execution model
Owner executes requests
Returns need-to-know
Remote Execution

Conformance of Access Control

Natural Language

2nd-Party Program Access Control

Satisfiability Modulo Theories (SMT)
Conformance Algorithm

- SMT: Generalization of boolean satisfiability (SAT) with theories of strings, arrays, ...
- Provably correct algorithms for conformance, and synthesis of conforming code
- NP-hard, but fast enough in practice

SMT (Satisfiability Modulo Theories)

Dad can monitor my security camera only if I am not home

\[
\sigma = @dad : \\
\text{monitor } @\text{security\_camera.event, } \\
\text{@phone.get\_gps()\{location }\neq \text{ home\} } \\
\Rightarrow \text{return}
\]

ThingTalk \((c)\)

\[
\sigma = @dad \\
\land (Y_1, \text{picture\_url}, Y_1, \text{has\_motion}, Y_1, \text{has\_person}) \\
= F_{\text{security\_camera.event}()} \\
\land (Y_2, \text{location}) = F_{\text{phone.get\_gps}()} \\
\land \text{home }\neq Y_2, \text{location}
\]

SMT Formula \((L[c])\)
Conformance of Access Controls

Conformance

\[ p \text{ conforms to } c \text{ if } p \preceq c \]
\[ \equiv L[p] \sqsubseteq L[c] \]
\[ \equiv \neg \text{SAT}(L[p] \land \neg L[c]) \]

Synthesis of a conforming program

The program \( p' = p \land c \)

is the least restrictive conforming program, provided \( p' \neq \text{null} \).

SMT is Fast Enough

- CVC4 SMT checker, v1.5
- 2.5 GHz Intel Xeon CPU, 80 GB RAM
- 50 policies allowing same functions run in 0.4 seconds
Needs and Acceptance?

Do Consumers Need Access Control?

<table>
<thead>
<tr>
<th>Role-Based Permission</th>
<th>Attribute-Based Permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teenage daughter to use credit card (Visa)</td>
<td>With a $20 budget limit</td>
</tr>
<tr>
<td>Amazon courier to unlock door</td>
<td>For restaurants only</td>
</tr>
<tr>
<td>If the package is over $1000</td>
<td>If your security camera is on</td>
</tr>
<tr>
<td>Friends to access cloud drive</td>
<td>Photos with their faces in them</td>
</tr>
<tr>
<td>Photos in a specific folder</td>
<td>Parent/kid to see security cameras</td>
</tr>
<tr>
<td>If you are not at home</td>
<td>Cameras facing the front yard/garage</td>
</tr>
<tr>
<td>10-year-old kid to use Netflix (Netflix)</td>
<td>Between 7 PM to 9 PM</td>
</tr>
<tr>
<td>Free G or PG rated movies</td>
<td>% People comfortable in giving permission (200 person survey)</td>
</tr>
</tbody>
</table>
More Examples

Willingness to share doubles with attribute-based access control

Expressiveness of ThingTalk?

Solicit use cases by showing AMT workers 3 examples, without describing ThingTalk

**Enforceable:**
Mom: “You need to follow this guy on Twitter, give me your Twitter account”.
Me: “OK, add him but don’t follow any other Twitter user”.

**Unenforceable:**
Friend: “Can I use your library card?”
Me: “OK, only if you return the book on time”.

---

CS243: Natural Language Programming

29

M. LAM

---

CS243: Natural Language Programming

30

M. LAM
ThingTalk is Expressive

60 people; 220 suggestions; 85 unique assets

Diverse use cases

- Business Accounts: 1%
- Social Media: 16%
- Services: 21%
- IoT: 27%
- Personal Data: 34%

85% in the scope of ThingTalk

- Unenforceable: 6%
- Out of Scope: 9%
- New API: 15%
- Existing API: 70%

User Study: “Sharing Without Passwords”

Like the concept? Like the app? Use the app?

<table>
<thead>
<tr>
<th>#users</th>
<th>12</th>
<th>10</th>
<th>8</th>
<th>6</th>
<th>4</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratings</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
6. Automatic Generation of GUI

- Linguistic user interface (LUI) is inadequate
- Future: LUI + GUI

Automatic GUI Generation

Fischer, Giovanni, Xu, Lam, MobileHCI 2018
Restyling in Real-Time

“monitor the camera in the baby room”
“monitor mails from my dad”
“adjust the volume of my speakers”
“translate ‘hello’ to Chinese”
“I’d like to order a coffee”

Bicentennial Print
Roy Lichtenstein, 1975
Summary

- Natural language programming
- Democratize programming: autonomy & privacy
- Will there be just a few formal constructs?
- Genie: a “compiler generator” to accelerate growth of linguistic capabilities
- A LUInet that understands HCI by 2024?

CS 294S: Virtual Assistants & Linguistic User Interfaces

- A first-experience research project course
- ML, programming+distributed systems, HCI, security
Example Projects

- **NLP:**
  - NL dialogs to define ThingTalk programs incrementally
  - Word embedding for linguistic interface vocabulary
  - Natural languages other than English
- **PL:**
  - Genie—improvement + new ThingTalk constructs
  - A LUI-GUI agnostic programming language
- **HCI:**
  - Novel, friendlier interfaces for virtual assistants
  - Mixed mode user interface: LUI + GUI
  - Knowledge engineering for virtual assistants:
    - Scraping the linguistic web from the graphical web
    - Templates for GDPR data released to users
- **Applications:**
  - A student’s virtual assistant
  - A car / hotel’s virtual assistant
  - A fitness assistant
  - A cardiologist’s virtual assistant (Prof. Allan Yeung)
  - Armchair conductor
    - Compose music in words & gestures (Prof. Ge Wang)
  - MedXChange (Profs. Ed Chang, David Mazieres, Lei Xing):
    - User-controlled 3rd party sharing
    - Human-understandable contracts
    - Scalable blockchain technology