An Intelligent Assistant for High-Level Task Understanding

Ming Sun

Yun-Nung Chen

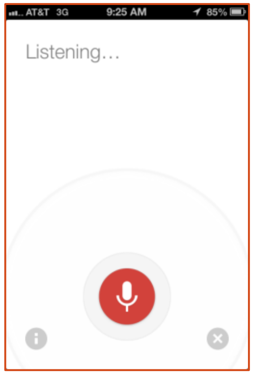
Alexander I. Rudnicky

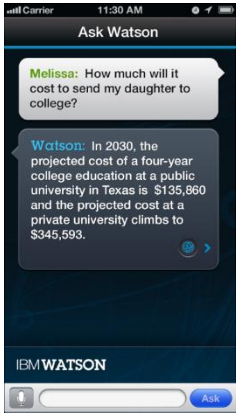
School of Computer Science

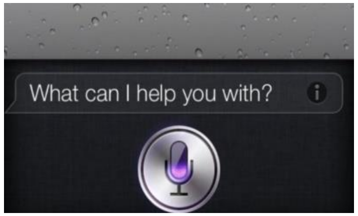
Carnegie Mellon University

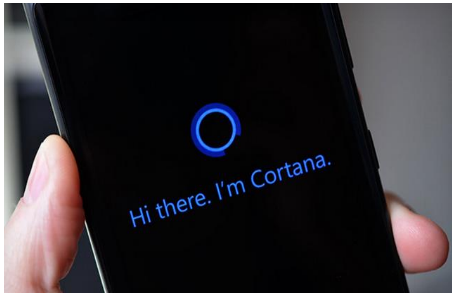
Many devices, many apps…

- Rich functionality is now available
- How well can it support user activities?

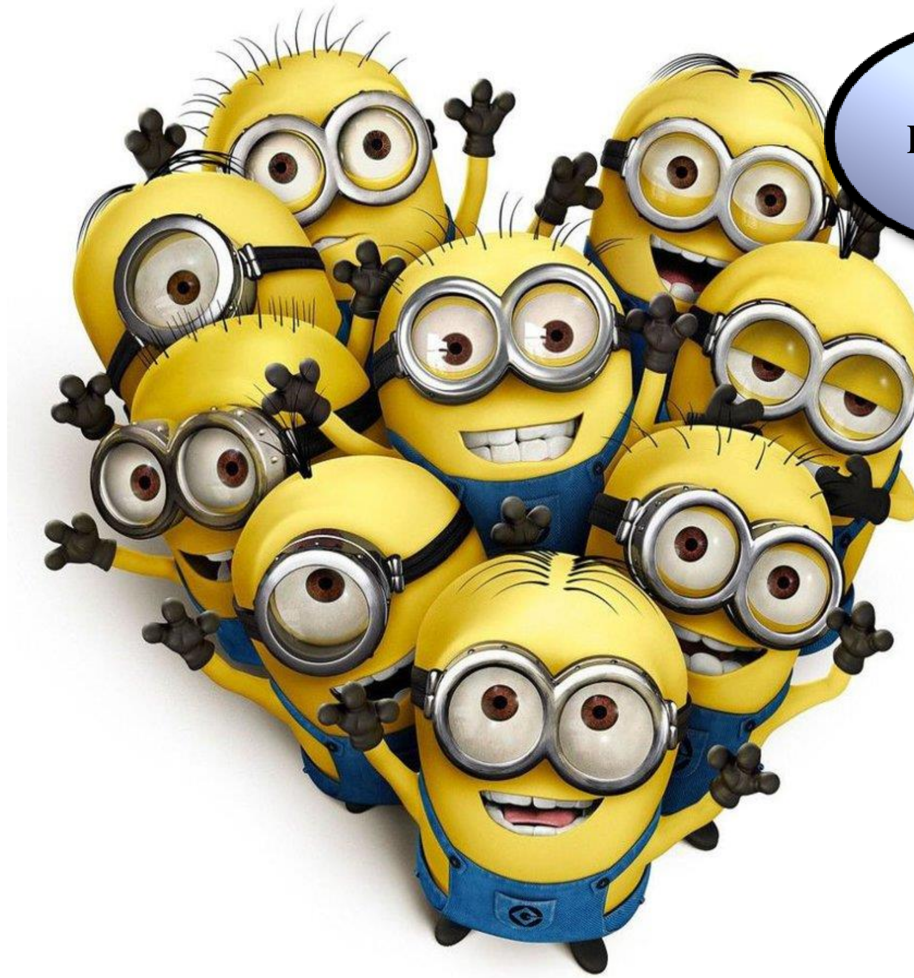








Book a restaurant!



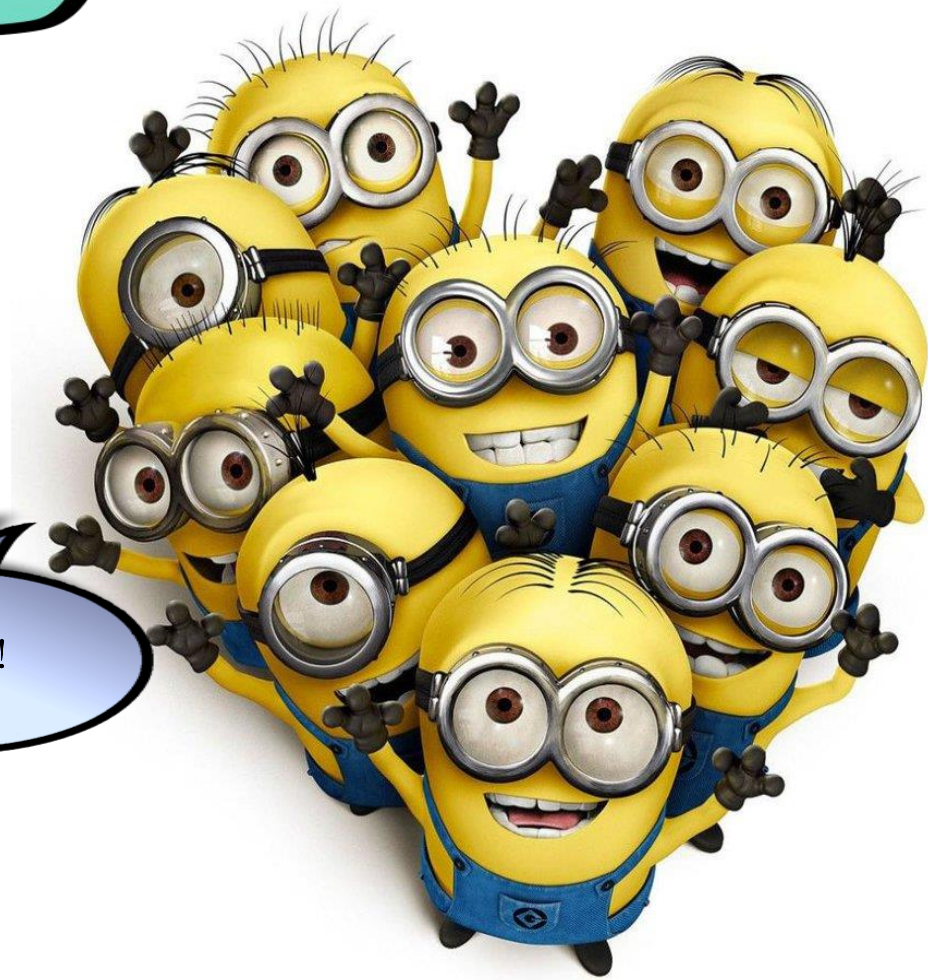
I can do **Yelp**!

Show me bus to China Palace.

I know **Maps**!

Tell kids we will have Chinese tonight.

I will **Text**!



Multi-domain Dialog – Current System

System: What can I do for you?

Cannot handle complex intention

User: Could you arrange a dinner for me and my friends?

System: Sorry I don't understand that. What can I do for you?

User: Can I book a table for three in Täkō downtown for this Friday?

…

System: OK. What can I do for you next?

Passively support cross-domain dialog

User: Show me the bus from here.

System: Where is your destination please?

No shared context

User: Täkō downtown Pittsburgh.

…

System: What should I do next?

User: Send the bus route to Carrie and Peter.

System: OK. Sending the following message to Carrie and Peter: "the bus route".

Multi-domain Dialog – Human Assistant

Understand complex intentions

Assistant: What can I do for you?

User: Could you arrange a dinner for me and my friends?

Assistant : What kind of food do you prefer?

User: Mexican?

Assistant : How about Täkō? I can book a table for you.

User: Sounds good! Can I take a bus there?

Actively support cross-domain dialog

Assistant : 61 A/B/C/D can take you there. Do you want to send this to your friends?

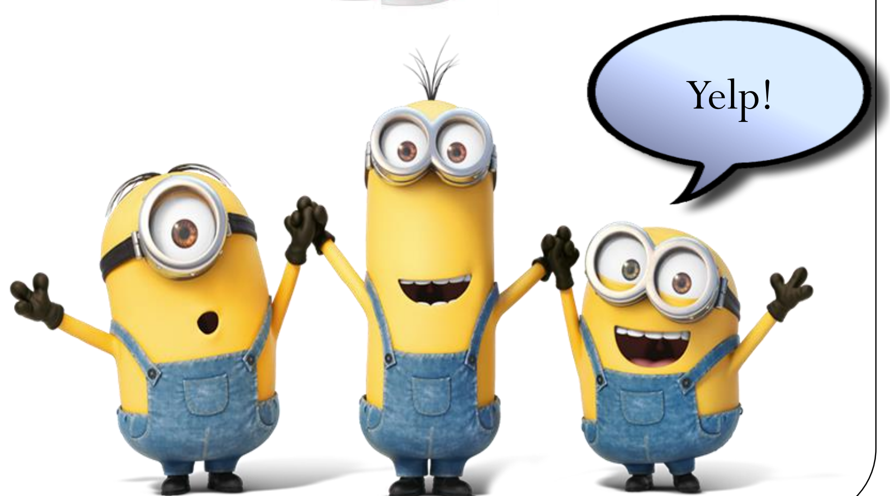
Maintain context

User: Great! Send it to Carrie and Peter.

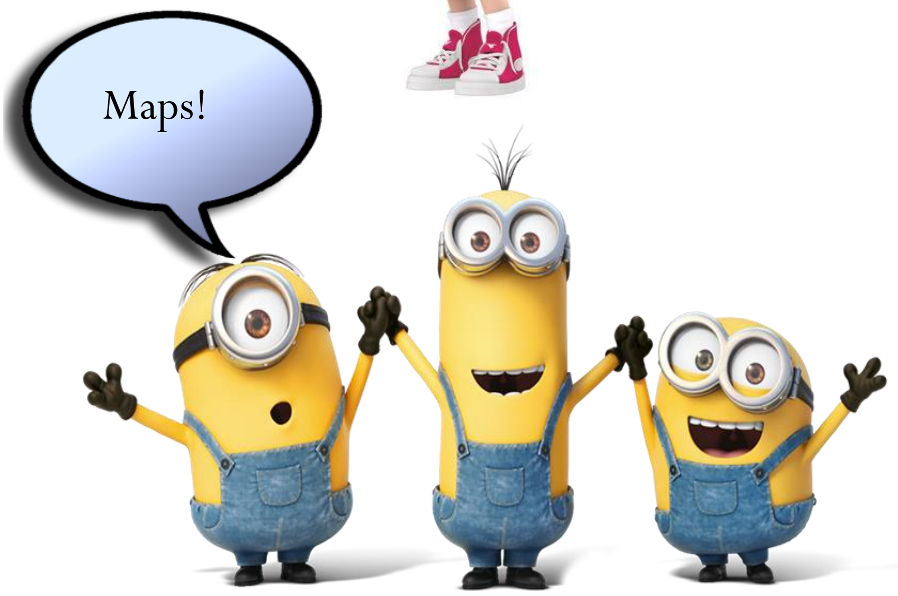
Assistant: OK. The bus route 61 has been sent.

Intention Understanding

Intention Understanding



Intention Understanding



Intention Understanding

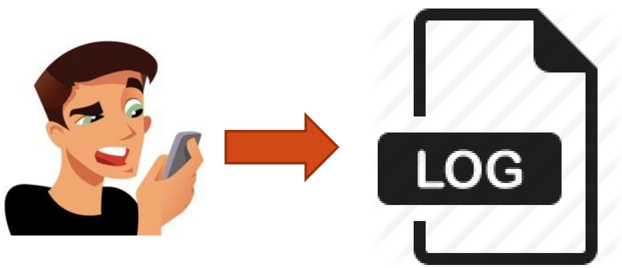
Intention Understanding

Approach

- Step 1: Observe human user perform multi-domain tasks

- Step 2: Learn to assist at task level
  - Map an activity description to a set of domain apps
  - Interact at the task level

Data Collection 1 – Smart Phone



- Log app invocation + time/date/location
- Separate log into episodes if there is 3 minute inactivity

Schedule a visit to CMU Lab

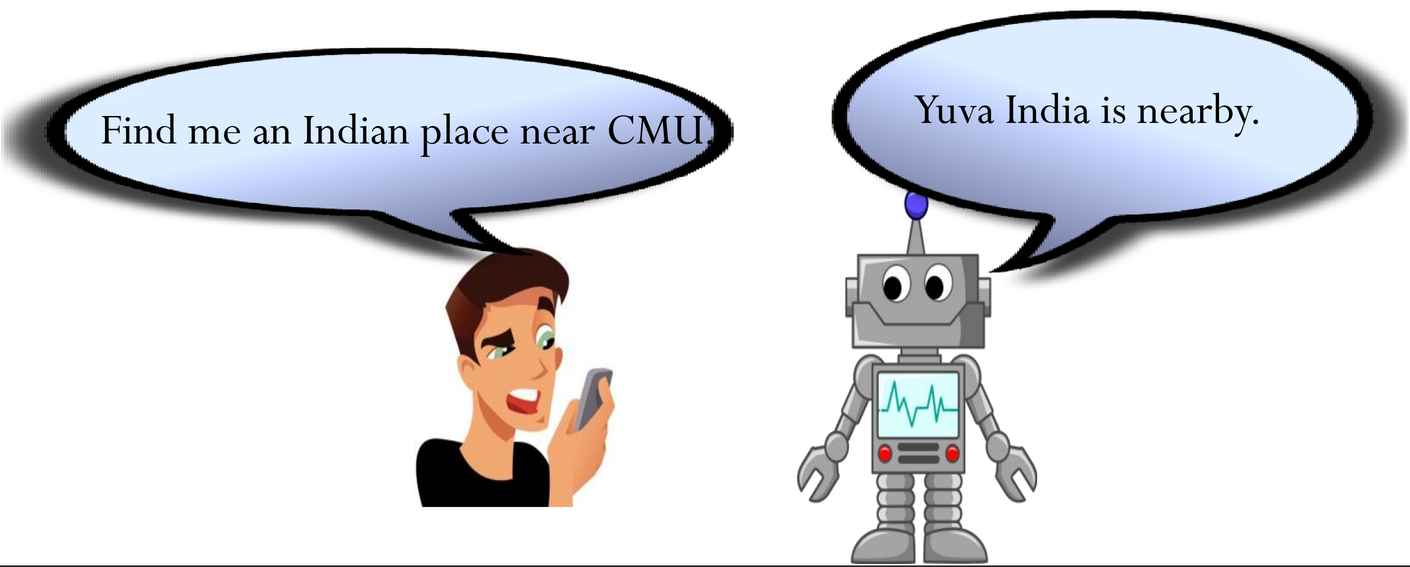
Wednesday 17:08 – 17:14 CMU	Messenger	Gmail	Browser	Calendar

Data Collection 2 – Wizard-of-Oz

Schedule a lunch with David.

Monday
10:08 – 10:15
Home

Yelp | Maps | Messenger | Music

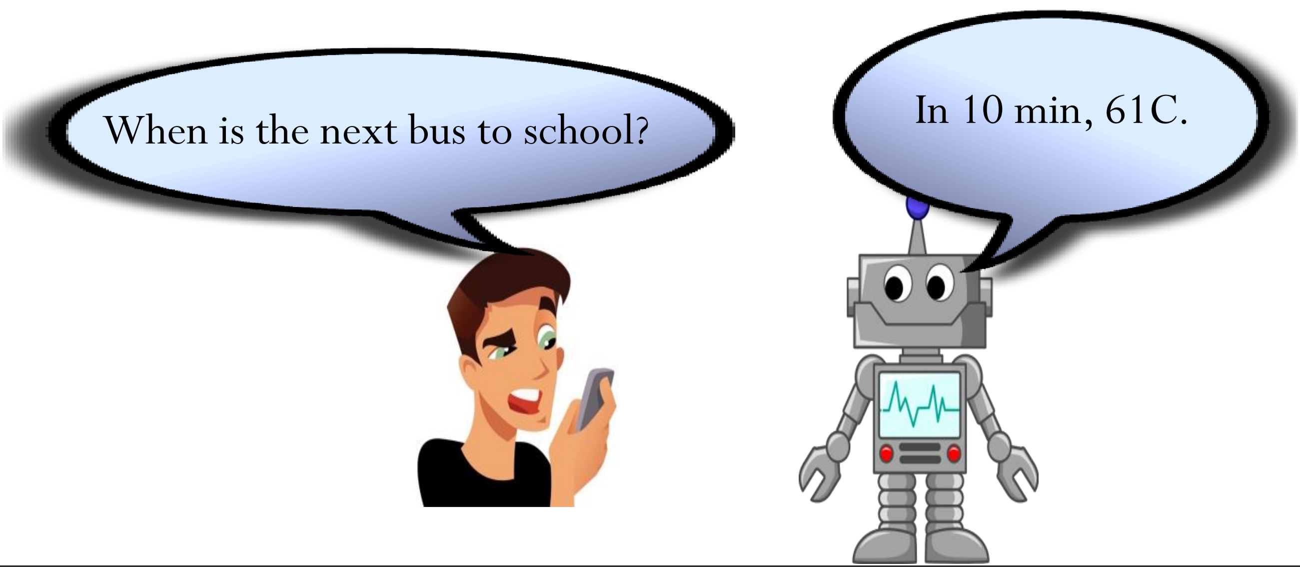


Data Collection 2 – Wizard-of-Oz

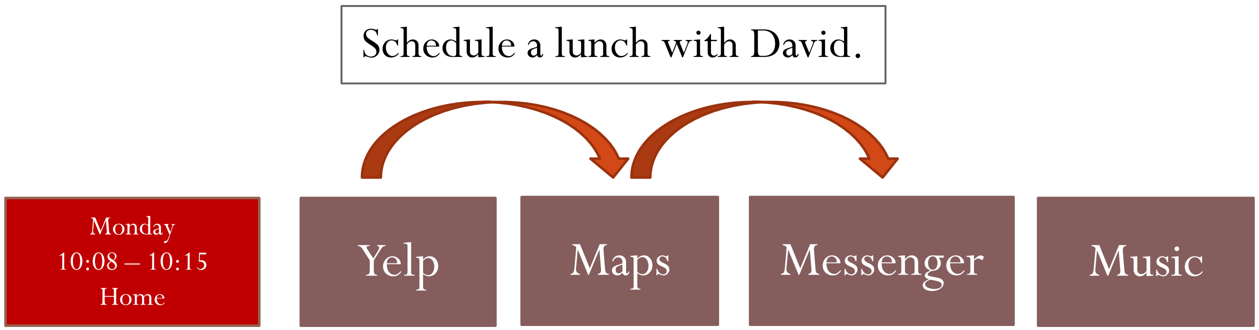
Schedule a lunch with David.

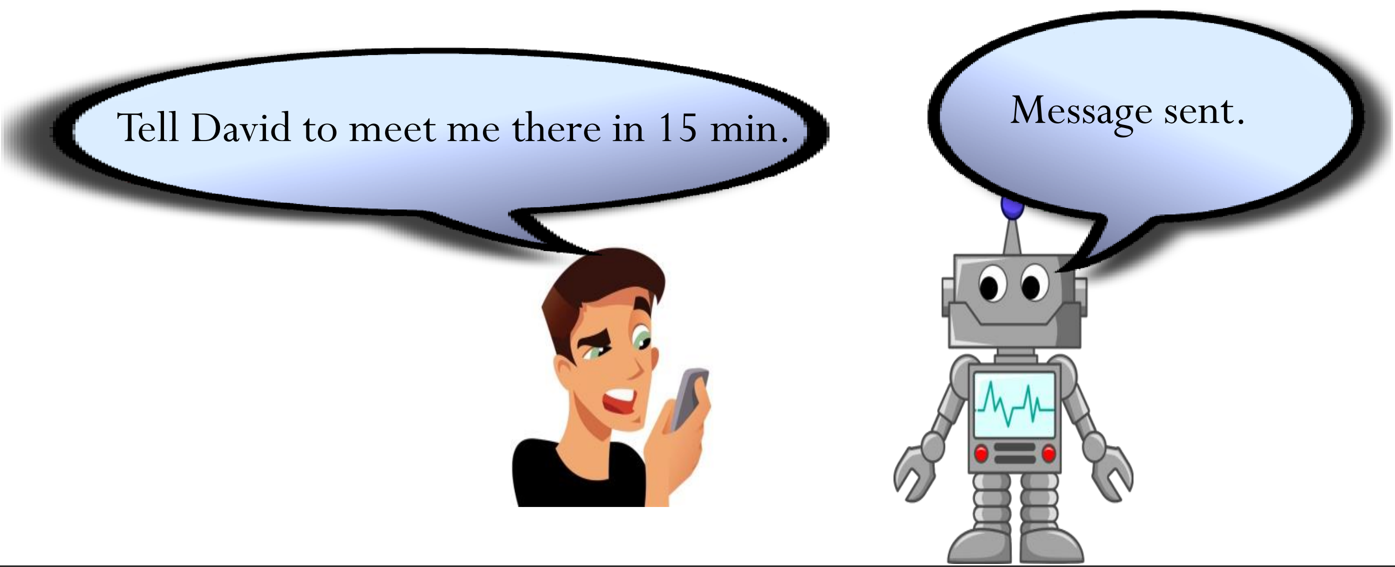
Monday
10:08 – 10:15
Home

Yelp | Maps | Messenger | Music



Data Collection 2 – Wizard-of-Oz

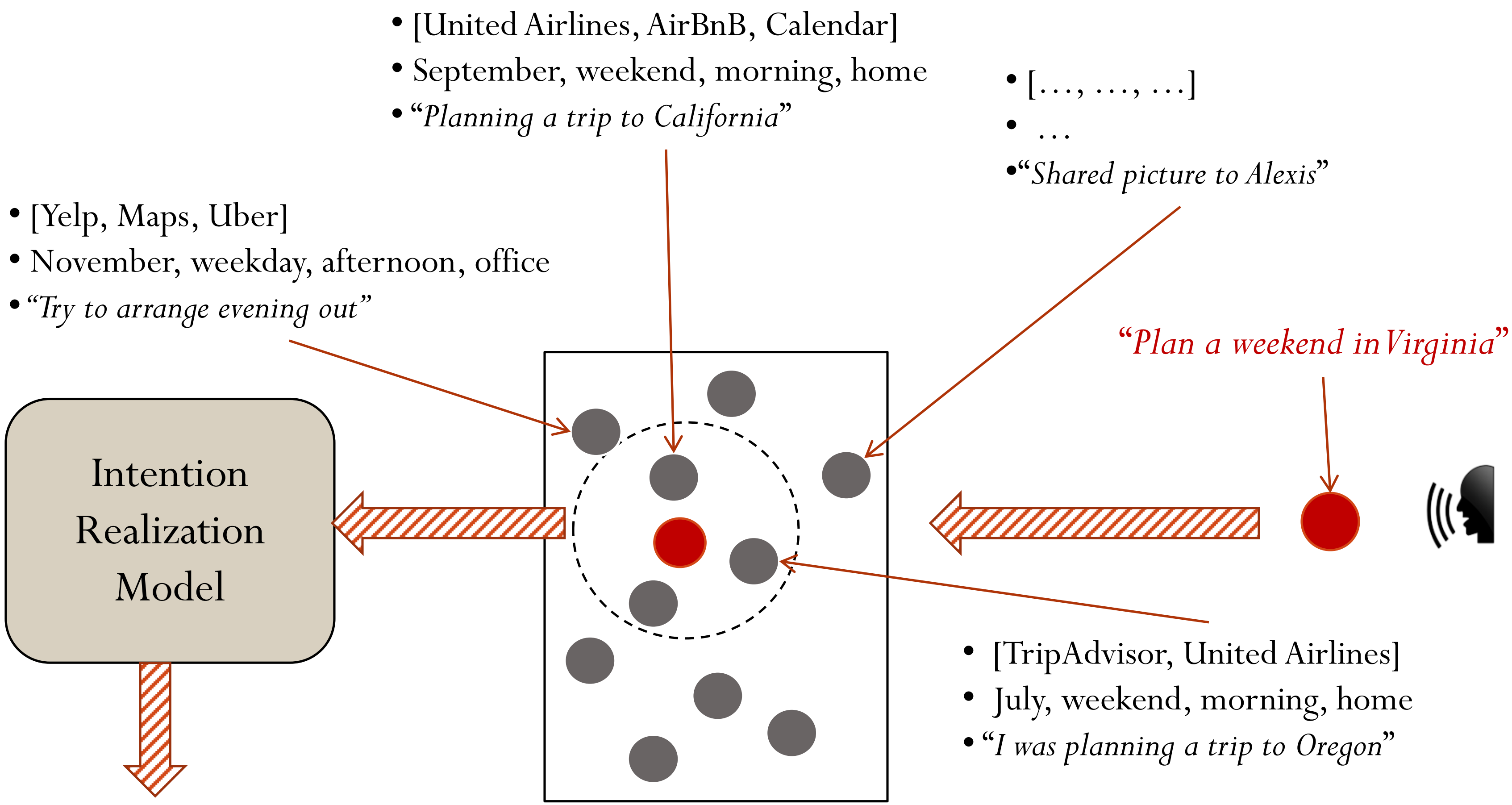




Corpus

- 533 real-life multi-domain interactions from 14 real users
  - 12 native English speakers (2 non-)
  - 4 males & 10 females
  - Mean age: 31
  - Total # unique apps: 130 (Mean = 19/user)

Resources	**Examples**	**Usage**
App sequences	Yelp->Maps->Messenger	structure/arrangement
Task descriptions	"Schedule a lunch with David"	nature of the intention, language reference
User utterances	"Find me an Indian place near CMU."	language reference
Meta data	Monday, 10:08 – 10:15, Home	contexts of the tasks

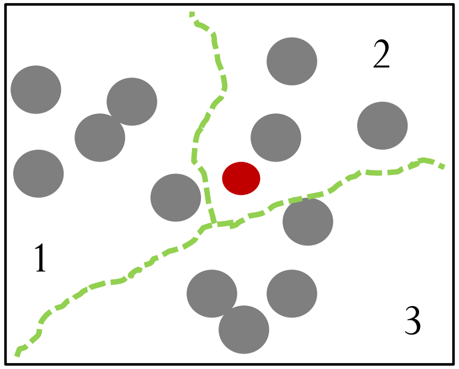


Infer:

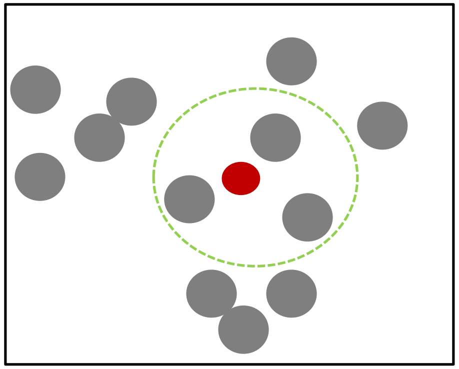
1) Supportive Domains: *United Airlines*, *TripAdvisor, AirBnB*
2) Summarization: "*plan trip*"

Find similar past experience

- Cluster-based:
  - K-means clustering on user generated language
- Neighbor-based:
  - KNN



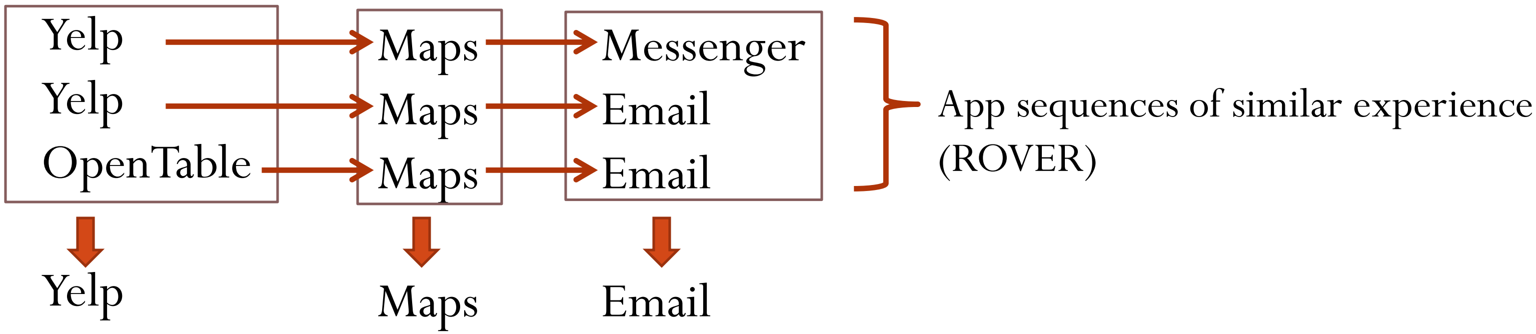
Cluster-based



Neighbor-based

Realize domains from past experience

- Representative Sequence



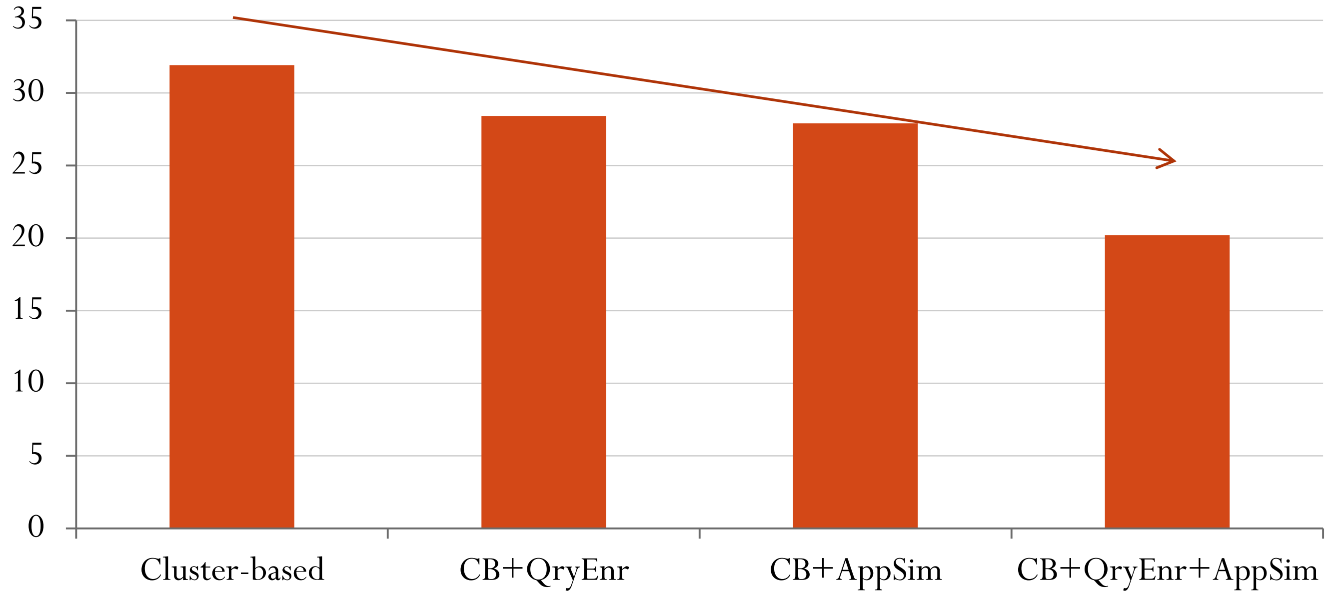
- Multi-label Classification

Some Obstacles to Remove

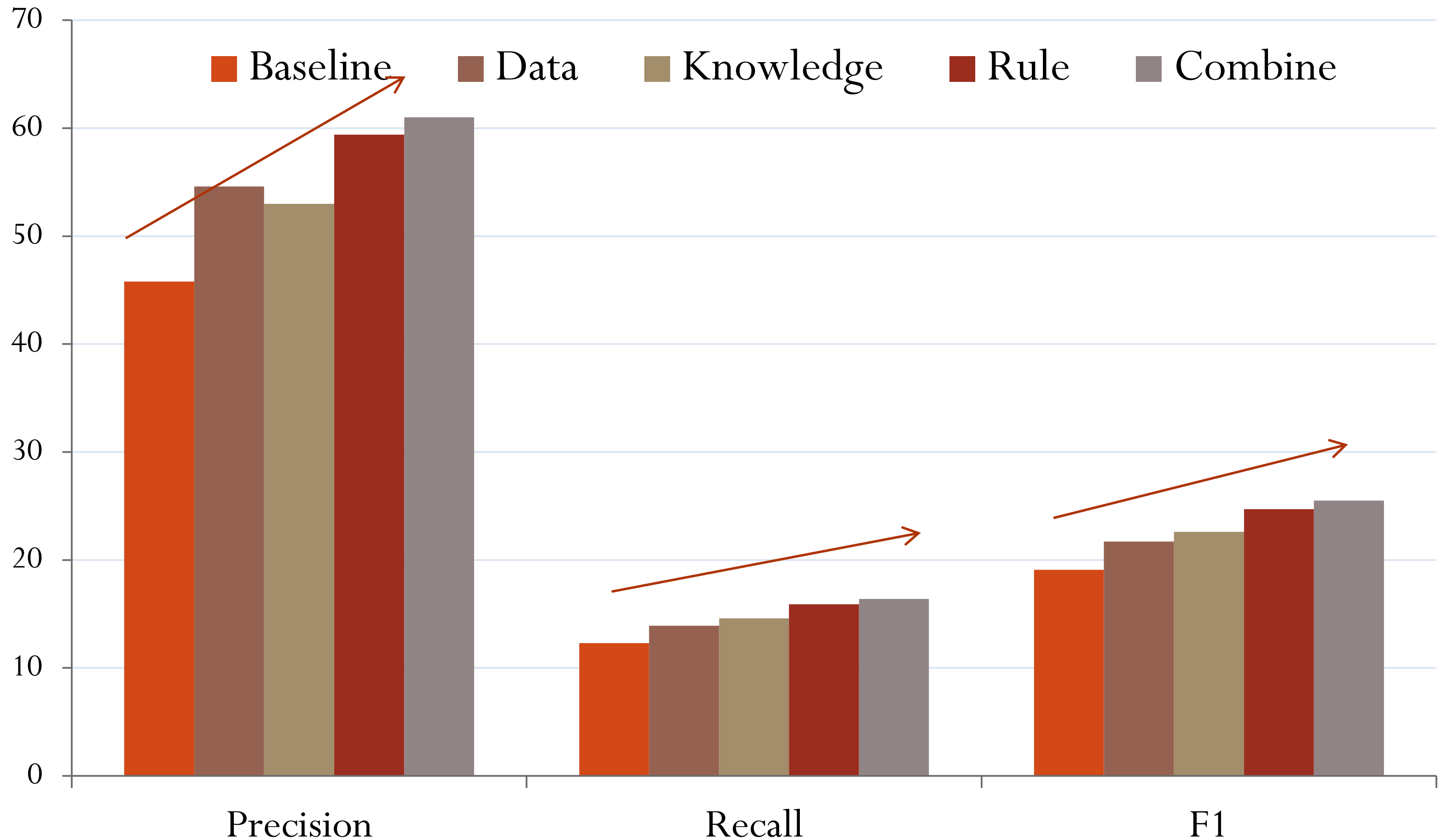
- Language-mismatch
  - **Solution**: Query Enrichment (QryEnr)
    - ["shoot", "photo"] -> ["shoot", "take", "photo", "picture"]
    - *word2vec, GoogleNews model*
- App-mismatch
  - **Solution**: App Similarity (AppSim)
    - Functionality space (derived from app descriptions) to identify apps
    - *Data-driven: doc2vec on app store texts*
    - *Rule-based: app package name*
    - *Knowledge-driven: Google Play similar app suggestions*

Gap between Generic and Personalized Models

QryEnr, AppSim, QryEnr+AppSim reduce the gap of F1



Compare different AppSim



Compare different AppSim

- Combining three approaches performs the best
- Knowledge-driven and data-driven have low coverage among (manufacture) apps
- Rule-based is better than the other two individual approaches

Learning to talk at the task level

- Techniques:
  - (Extractive/abstractive) summarization
  - Key phrase extraction [RAKE]

- User study:
  - Key phrase extraction + user generated language
  - Ranked list of key phrases + user's binary judgment

```
[descriptions]
Looking up math problems.
Now open a browser.
Go to slader.com.
Doing physics homework.
…
[utterances]
Go to my Google drive.
Look up kinematic equations.
Now open my calculator so I can plug in numbers.
…
```

```
1. solutions online
2. project file
3. Google drive
4. math problems
5. physics homework
6. answers online
…
```

Learning to talk at the task level

- Metrics
  - Mean Reciprocal Rank (MRR)

- Result:
  - MRR ~0.6
    - understandable verbal reference show up in top 2 of the ranked list

```
[descriptions]
Looking up math problems.
Now open a browser.
Go to slader.com.
Doing physics homework.
…
[utterances]
Go to my Google drive.
Look up kinematic equations.
Now open my calculator so I can plug in numbers.
…
```

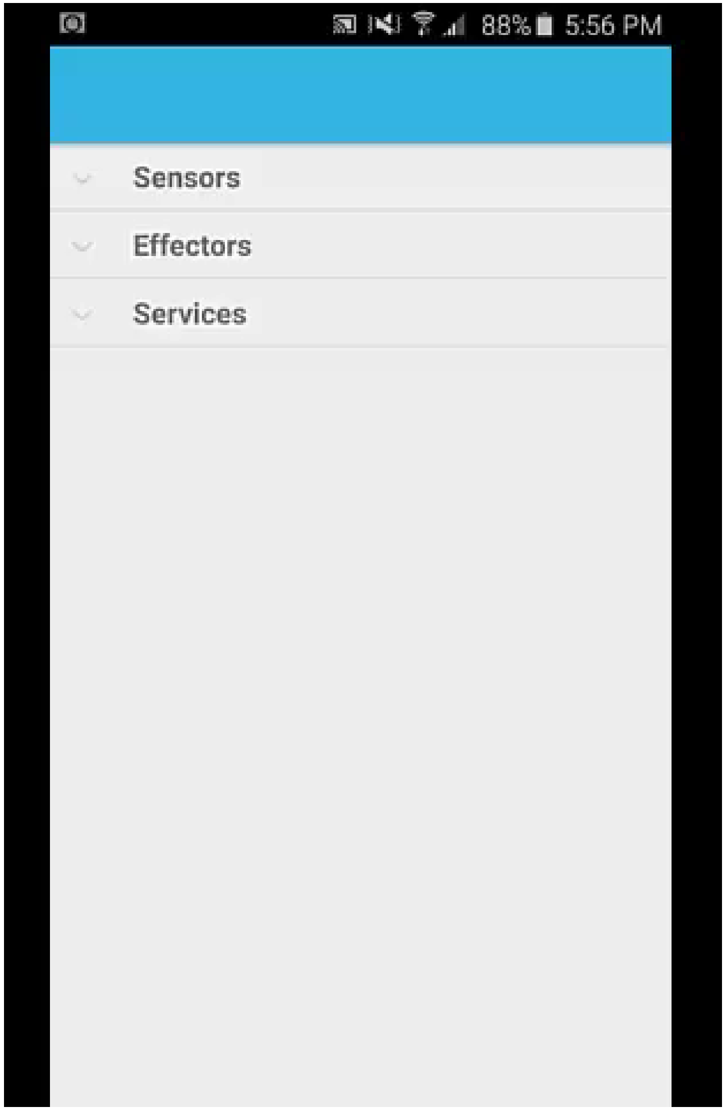
```
1. solutions online
2. project file
3. Google drive
4. math problems
5. physics homework
6. answers online
…
```

Summary

- Collected real-life cross-domain interactions from real users
- HELPR: a framework to learn assistance at the task level
  - Suggest a set of supportive domains to accomplish the task
    - Personalized model > Generic model
    - The gap can be reduced by QryEnr + AppSim
  - Generate language reference to communicate verbally at task level

HELPR demo

- Interface
  - HELPR display
  - GoogleASR
  - *Android TTS*
- HELPR server
  - User models



Thank you

- Questions?