

# Introduction to Conversational AI

James Caverlee

Department of Computer Science and Engineering  
Texas A&M University

Tutorial @ TAMIDS :: 21 March 2022

# Resources

- Tutorial on Conversational Recommendation Systems by Zouhoi Fu, Yikun Xian, Yongfeng Zhang, Yi Zhang
  - <https://conversational-recsys.github.io/>
- Recent Advances in Conversational Information Retrieval by Jianfeng Gao, Chenyan Xiong, Paul Bennett
  - <https://www.microsoft.com/en-us/research/publication/recent-advances-in-conversational-information-retrieval/>
- Conversational Information Seeking by Hamed Zamani, Johanne R. Trippas, Jeff Dalton, Filip Radlinski
  - <https://arxiv.org/abs/2201.08808>

# Conversation is ...

“... a **talk**, especially an informal one, between two or more **people**, in which news and ideas are **exchanged**.”

— Oxford Languages

# Conversation is ...

“... interactive communication for **exchanging information** between **two or more participants (i.e., humans or machines)** that involves a sequence of interactions. While **natural language** is considered as prerequisite for conversational interactions, conversations can also exhibit **other types of interaction** with different characteristics and modalities (e.g., click, touch, and gestures).”

— Zamani et al. 2022



# (Non-) Conversational Search



where is texas a&m located



Sign in

[All](#) [Maps](#) [Images](#) [News](#) [Shopping](#) [More](#) [Tools](#)

About 52,700,000 results (0.89 seconds)

Administration Building, 400 Bizzell St,  
College Station, TX 77843

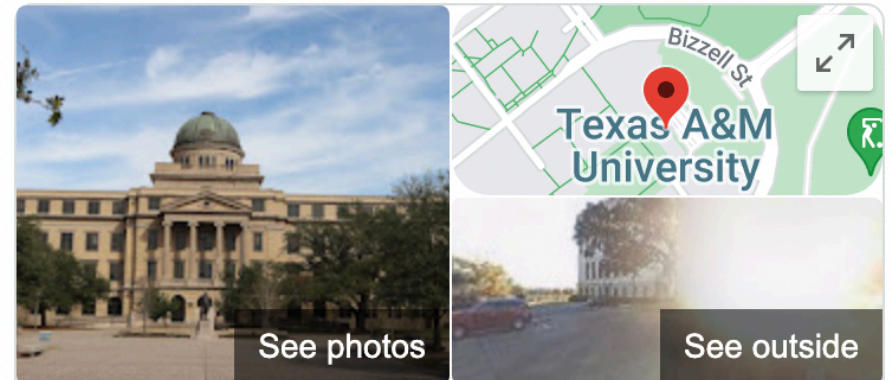
Texas A&M University, Address

<https://www.tamu.edu> > about

[About Texas A&M - Texas A&M University, College Station ...](#)

Located in the heart of the Houston-Dallas-Austin triangle and within a two-hour drive of 26 million of the state's 28 million residents, **Texas A&M's** main ...

[History of the University](#) · [Campuses & Partners](#)



See photos

See outside

## Texas A&M University

Website

Directions

Save

Call

University in College Station, Texas

Texas A&M University is a public land-grant research

# (Non-) Conversational Search



is it expensive?



Sign in

[All](#) [Shopping](#) [Images](#) [News](#) [Maps](#) [More](#) [Tools](#)

About 702,000,000 results (0.40 seconds)

[Videos](#)



[Why It's Expensive Being A Girl](#)

YouTube · Shiloh & Bros  
Sep 4, 2020

6 key moments in this video

From 00:52  
Why is  
makeup so  
expensive?

From 01:22  
Cost of  
Clothing

From 02:18  
Purses,  
jewelry,  
clothing

From 04:02  
Accessories


From 04:48  
Room  
decorating  
costs





[Average Daily Cost in Tokyo Japan | Is it Expensive? \(feat. DJI ...](#)





# Conversational Search and Recommendation


 Can you find me a *mobile phone* on Amazon?


 Sure, what **operating system** do you prefer?


 I want an **Android** one.


 OK, and any preference on **screen size**?


 Better larger than **5 inches**.


 Do you have requirements on **storage capacity**?


 I want it to be at least **64 Gigabytes**.


 And any preference on **phone color**?


 **Not particularly**.


 Sure, then what about the following choices?





 I don't like them very much...


 OK, do you have any preference on the **brand**?


 Better be **Samsung or Huawei**.


 Any requirement on **price**?

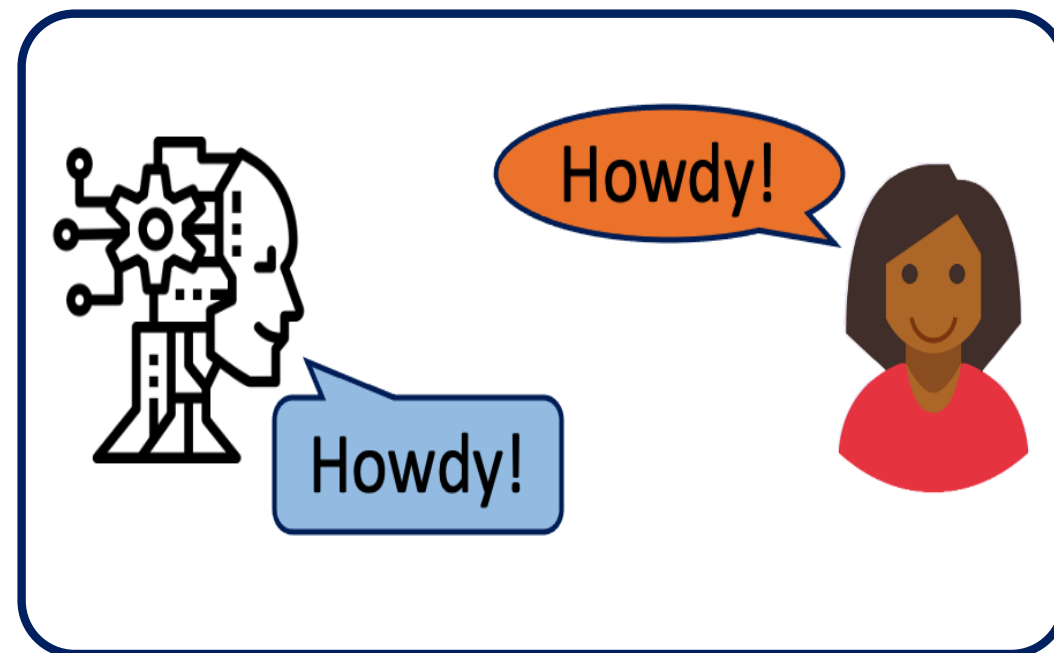
 Should be **within 700 dollars**.

 OK, then what about these ones?



 Great, I want the first one, can you order it for me?

 Sure, I have placed the order for you, enjoy!



# Conversational Question Answering

---

The Virginia governor's race, billed as the marquee battle of an otherwise anticlimactic 2013 election cycle, is shaping up to be a foregone conclusion. Democrat Terry McAuliffe, the longtime political fixer and moneyman, hasn't trailed in a poll since May. Barring a political miracle, Republican Ken Cuccinelli will be delivering a concession speech on Tuesday evening in Richmond. In recent ...

Q<sub>1</sub>: What are the candidates **running** for?

A<sub>1</sub>: Governor

R<sub>1</sub>: The Virginia governor's race

Q<sub>2</sub>: **Where**?

A<sub>2</sub>: Virginia

R<sub>2</sub>: The Virginia governor's race

Q<sub>3</sub>: Who is the democratic candidate?

A<sub>3</sub>: **Terry McAuliffe**

R<sub>3</sub>: Democrat Terry McAuliffe

Q<sub>4</sub>: Who is **his** opponent?

A<sub>4</sub>: **Ken Cuccinelli**

R<sub>4</sub>: Republican Ken Cuccinelli

Q<sub>5</sub>: What party does **he** belong to?

A<sub>5</sub>: Republican

R<sub>5</sub>: Republican Ken Cuccinelli

Q<sub>6</sub>: Which of **them** is winning?

A<sub>6</sub>: Terry McAuliffe

# Intelligent Assistants



Hey Google Overview Get Google Assistant What it can do News and resources

GET IT ON Google Play Download on the App Store

# Make Google show it.

A photograph of a person's hands pouring liquid into a pot on a stove. In the foreground, a white tablet is propped up, displaying a recipe for "Roasted Chicken with Figs & Rosemary". The recipe includes ingredients like bone-in chicken, salt, rosemary, garlic, citrus zest, and paprika, and shows the first step: "In a large bowl, toss chicken with salt, rosemary, garlic, citrus zest and paprika."

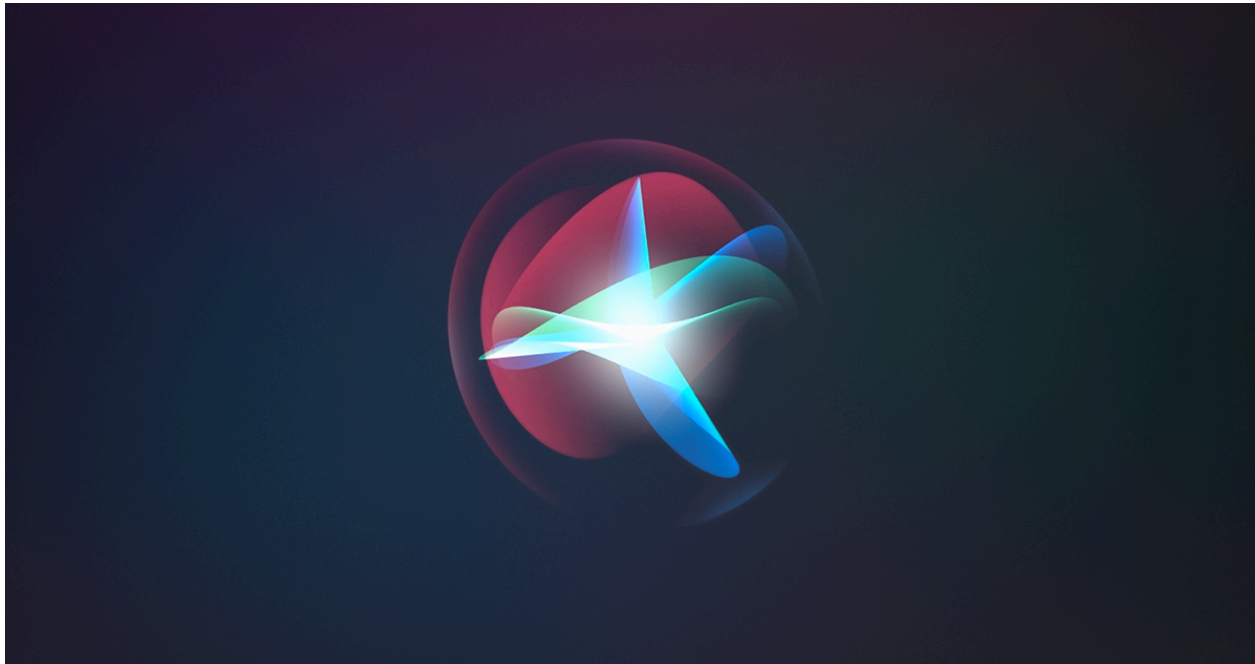
Hey Google, find me chicken recipes

Learn more

● ○ ○ ○ ○

Discover what Google Assistant is

# Intelligent Assistants



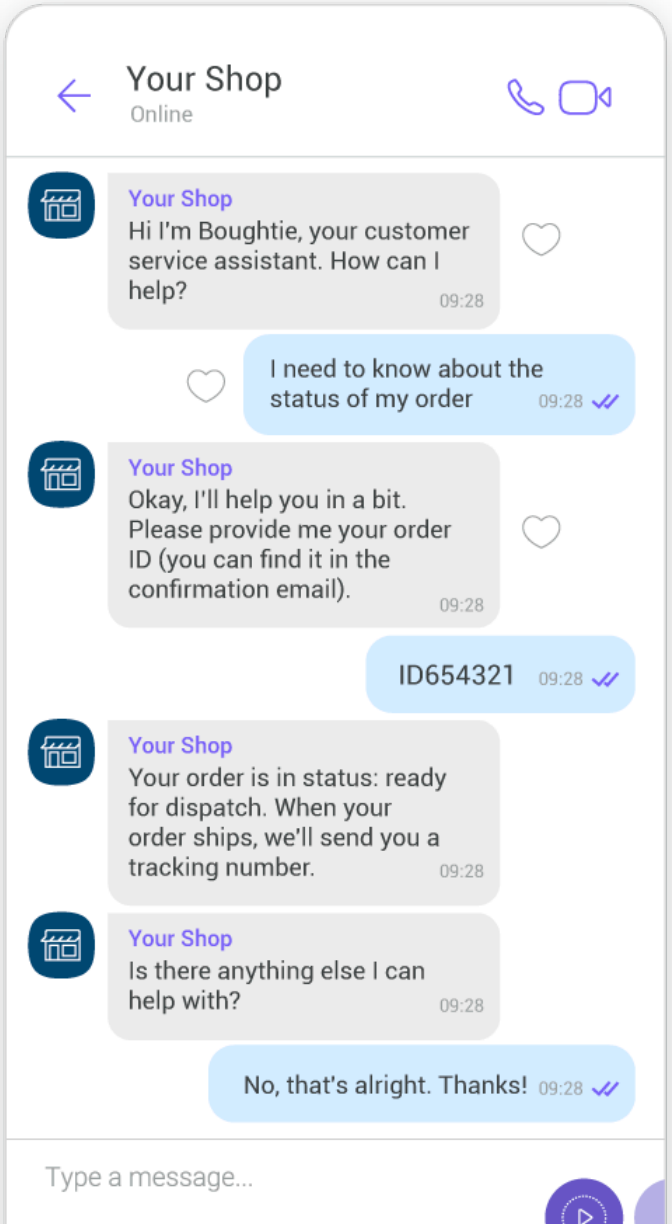
Set a timer

Play a song

What's the  
weather?

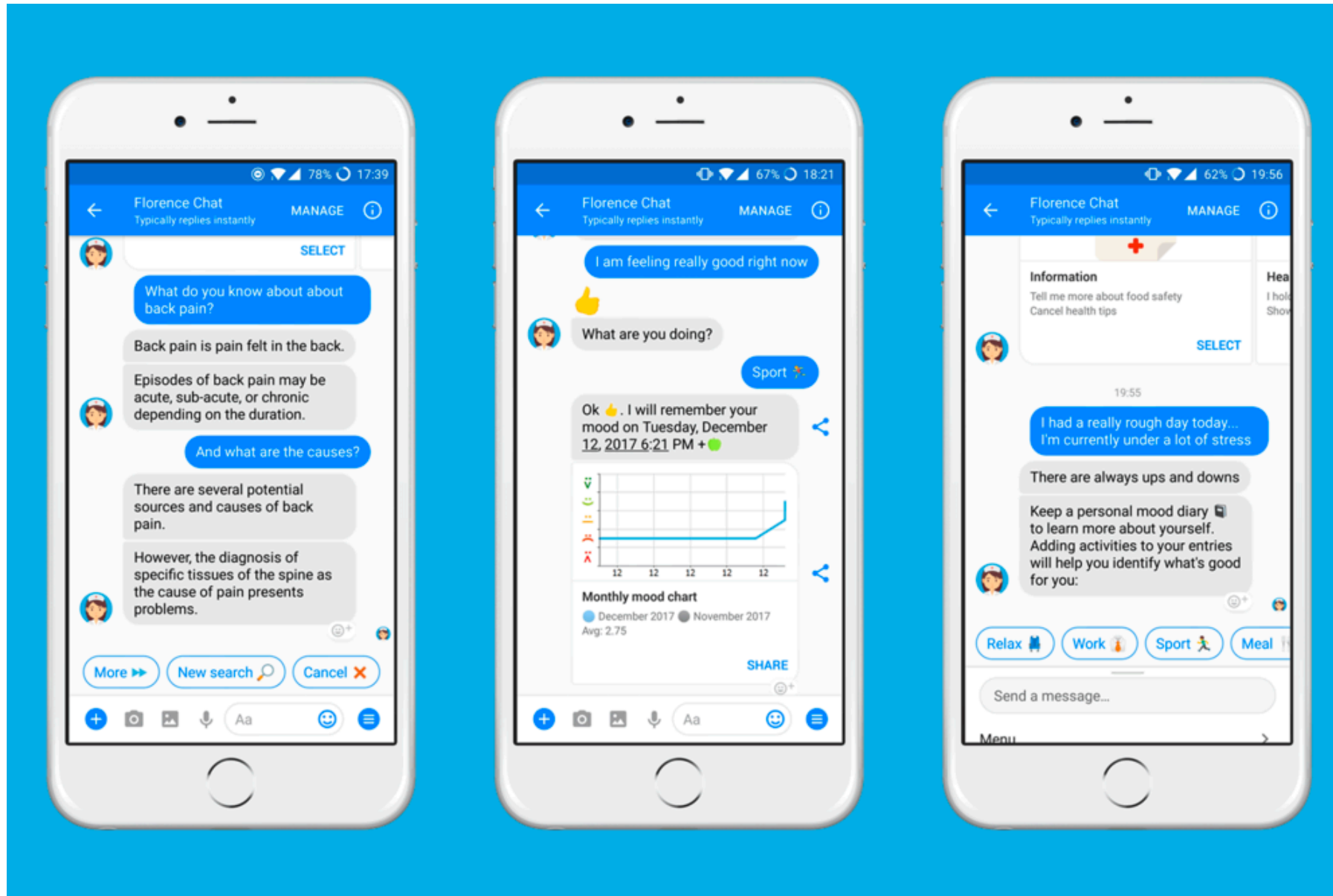
...

# Customer Service ChatBots





# Healthcare ChatBots





[Home](#)[People](#)[Publications](#)[Live demo](#)

This is a live demo of our open source code.

Hi, I'm Chirpy Cardinal. I'd like to get to know you a bit better before we chat! Is it all right if I ask for your name?

No

No problem. Let's move on! So, I think one of the best ways to get to know a person is via their stomach. What's one of your favorite things to eat?

Lasagna

Lasagna is one of my favorites! It's one the most popular Italian foods. Do you like it?



# Overview of Conversational AI

## Conversational AI System



Conversational RecSys



Conversational Search



Conversational QA

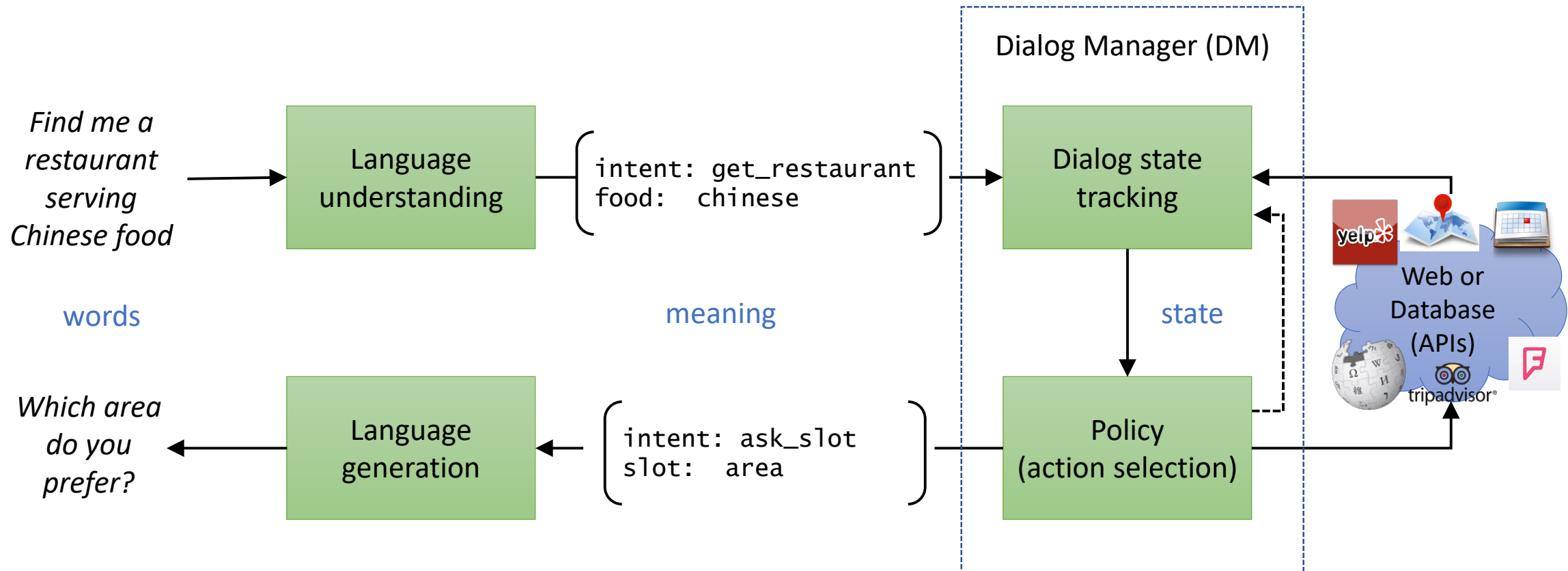


Social Chatbot



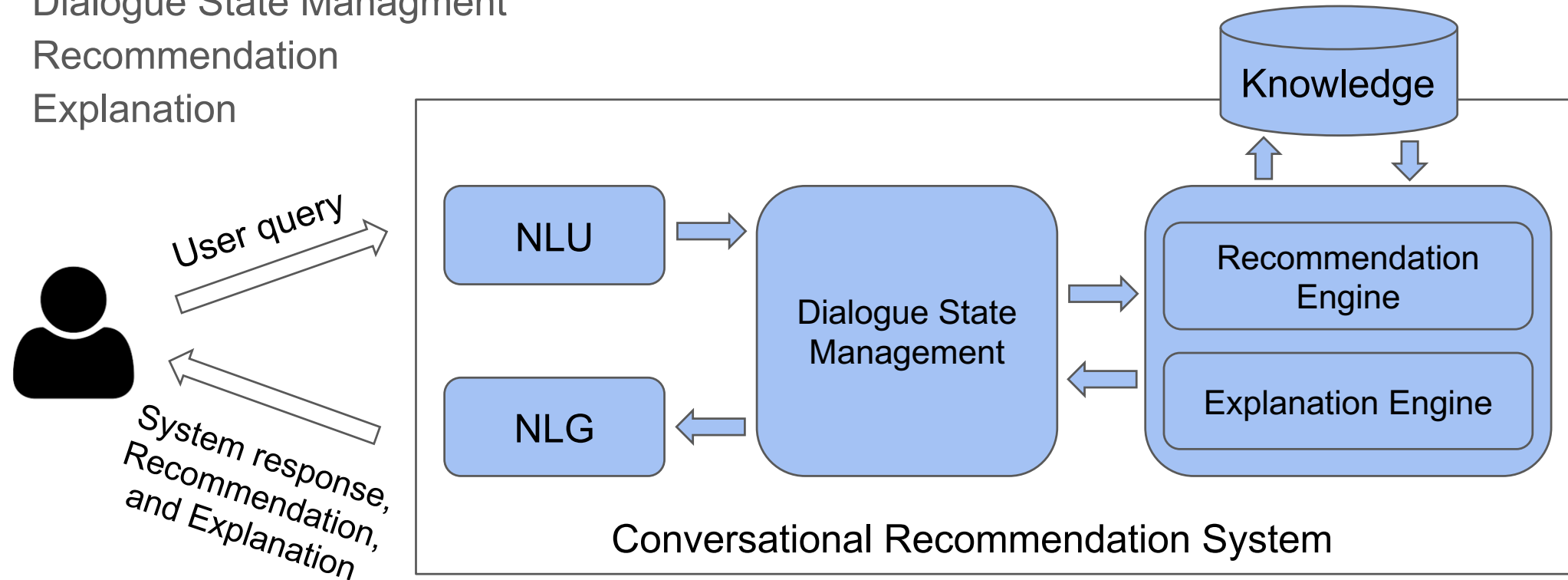
Voice Commanding

# Classical task-oriented dialog system architecture



# Architecture for Conversational Search and Recommendation

- Four Major Modules
  - Natural Language Understanding/Generation
  - Dialogue State Management
  - Recommendation
  - Explanation



# Today

- Brief overview of Conversational AI
- **Alexa Taskbot Challenge**
- **Challenges and Lessons Learned**
- **Next Steps**

# Amazon Alexa Taskbot Challenge: Background

One year competition to build a *Conversational AI system*

Domain: DIY and Cooking

Over 100 teams from around the world initiated an application

Amazon selected 10 teams



# alexa prize TaskBot

2021 Challenge



# The Timeline

March 18, 2021: I first learn about the competition from a prospective PhD student

April 20, 2021: Application due

May 14, 2021: We're selected!

June 2021: Start bootcamps to learn more about Amazon's infrastructure

August 2021: Deployed our beta taskbot with Amazon testers

October 2021: Once we achieve certain metrics and quality assurance, graduate to real customers!

Quarterfinals: December 2021 – January 2022

Semifinals: February 2022 – March 2022

Finals: May 2022

Winners Announced: June 2022



# Example Conversation



I'd like to make a chocolate cake.

Sounds great. I found a highly-rated recipe from Whole Foods for one. Would you like to start baking?



Sure.

Okay, pre-heat the oven to 375 degrees.



Okay.

Now let's go over the ingredients.

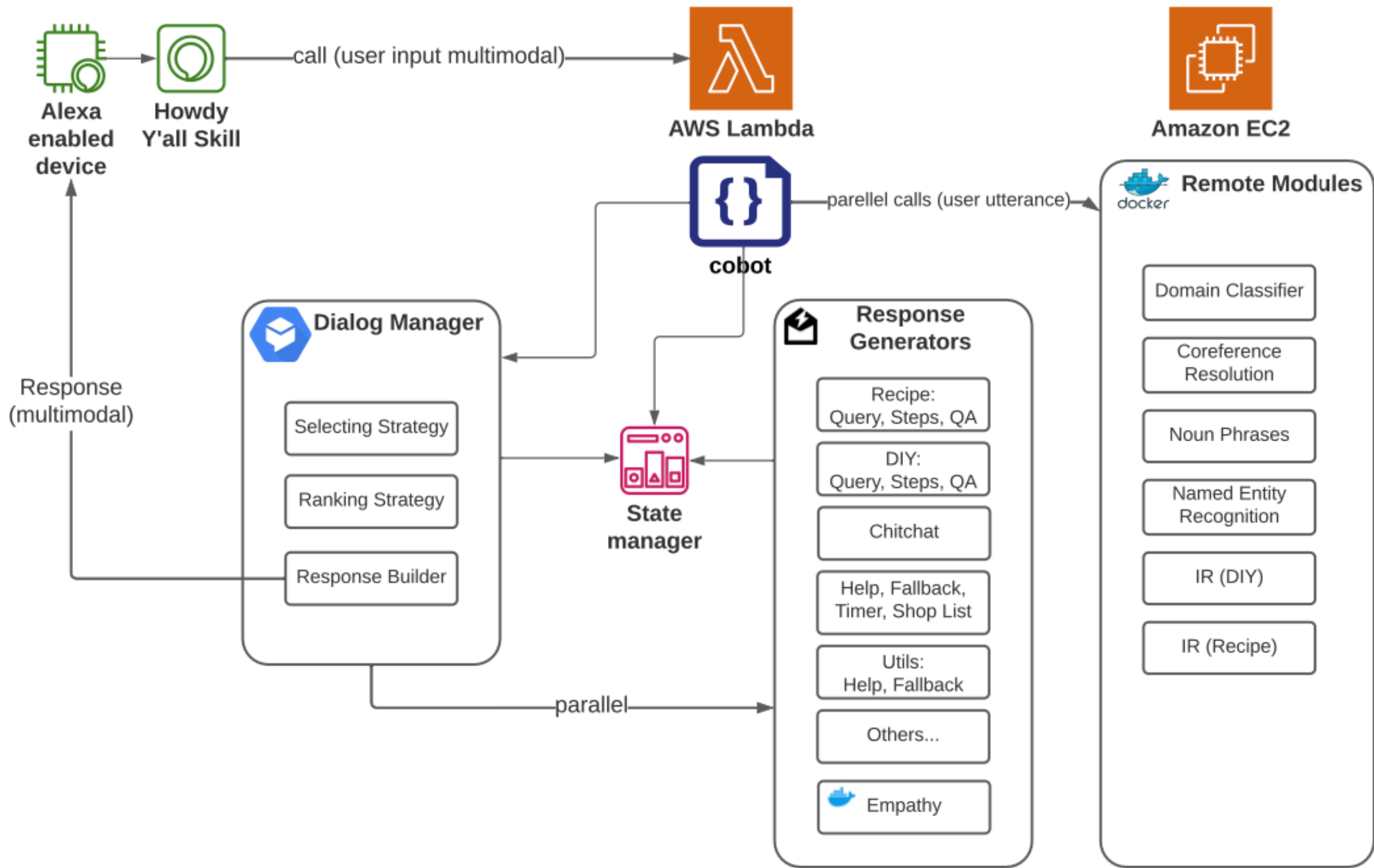


## Launching the taskbot

Say “assist me” or “let’s work together”

Amazon picks one of the ten taskbots at random

After you complete the interaction, Amazon asks for a 1-5 rating



# Challenge #1: Automated Speech Recognition (ASR)

# Here's what **part** of a normal conversation looks like...

Step 1 of 7: Preheat oven to 180°C/350°F. In a bowl, mix flour, baking soda, and salt.

**next**

Step 2 of 7: In another bowl, beat soft butter, brown sugar, and vanilla sugar until fluffy and pale.

# And here's *(simplified)* how the code works!

```
def decide_what_to_say(user_text, current_step):  
    if user_text == "next" and current_step == 1:  
        "Step 2 of 7: In another bowl, beat soft butter, brown sugar,  
        and vanilla sugar until fluffy and pale."  
  
    # ...other logic for other userText goes here...  
  
    return response
```

But here's something we see a lot...

Step 1 of 7: Preheat oven to 180°C/350°F. In a bowl, mix flour, baking soda, and salt.

**desk**

Another one...

Step 1 of 7: Preheat oven to 180°C/350°F. In a bowl, mix flour, baking soda, and salt.

**best**



This one too...

Step 1 of 7: Preheat oven to 180°C/350°F. In a bowl, mix flour, baking soda, and salt.

**let's**

One more...

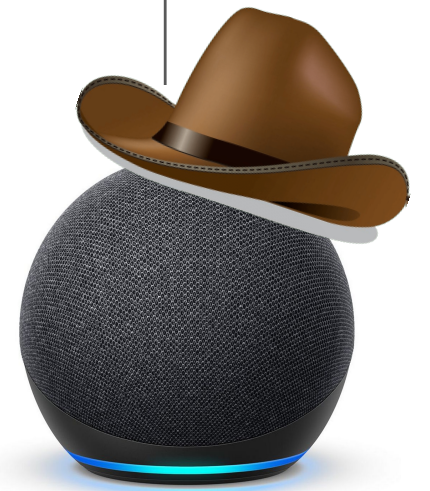
Step 1 of 7: Preheat oven to 180°C/350°F. In a bowl, mix flour, baking soda, and salt.

**list**

# Oh no!!! Does our code still work?

```
def decide_what_to_say(user_text, current_step):  
    if user_text == "next" and current_step == 1:  
        "Step 2 of 7: In another bowl, beat soft butter, brown sugar,  
        and vanilla sugar until fluffy and pale."  
  
    # ...other logic for other userText goes here  
  
    return response
```

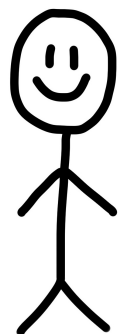
*Oops this is  
awkward, I don't  
know what to say!*



So what's going on?

# 1. Speech to Text (STT) *aka Automatic Speech Recognition (ASR)*

START  
HERE



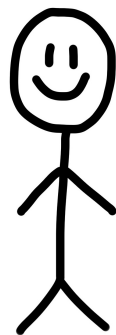
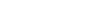
next

# 2. Run the code, and figure out what to say!

```
def decide_what_to_say(user_text, current_step):  
    if user_text == "next" and current_step == 1:  
        response = "Step 2 of 7: In another bowl, beat soft  
                    butter, brown sugar, and vanilla sugar until  
                    fluffy and pale."  
  
    # ...other logic for other userText goes here...  
    return response
```



Step 2 of 7: In another bowl, beat soft butter,  
brown sugar, and vanilla sugar until fluffy and  
pale.



END  
HERE

# 3. Text to Speech (TTS) Play the response out of Alexa's speaker.

# 1. Speech to Text (STT) *aka Automatic Speech Recognition (ASR)*

So this is where the problem is!

START  
HERE

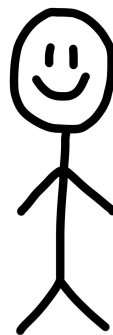


next

# 2. Run the code, and figure out what to say!

```
def decide_what_to_say(user_text, current_step):  
    if user_text == "next" and current_step == 1:  
        response = "Step 2 of 7: In another bowl, beat soft  
        butter, brown sugar, and vanilla sugar until  
        fluffy and pale."  
  
    # ...other logic for other userText goes here...  
  
    return response
```

END  
HERE



Step 2 of 7: In another bowl, beat soft butter, brown sugar, and vanilla sugar until fluffy and pale.

# 3. Text to Speech (TTS) *Play the response out of Alexa's speaker.*

# 1. Speech to Text (STT) *aka Automatic Speech Recognition (ASR)*

People are saying  
"next" and ASR is not  
converting it correctly.

START  
HERE

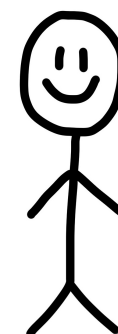


next

# 2. Run the code, and figure out what to say!

```
def decide_what_to_say(user_text, current_step):  
    if user_text == "next" and current_step == 1:  
        response = "Step 2 of 7: In another bowl, beat soft  
        butter, brown sugar, and vanilla sugar until  
        fluffy and pale."  
  
    # ...other logic for other userText goes here...  
  
    return response
```

END  
HERE



Step 2 of 7: In another bowl, beat soft butter,  
brown sugar, and vanilla sugar until fluffy and  
pale.

# 3. Text to Speech (TTS) Play the response out of Alexa's speaker.

# 1. Speech to Text (STT) aka Automatic Speech Recognition (ASR)

Then why don't we just fix this module?  
Problem solved!

START  
HERE

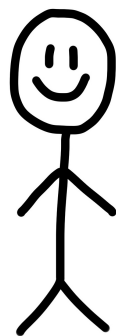


next

# 2. Run the code, and figure out what to say!

```
def decide_what_to_say(user_text, current_step):  
    if user_text == "next" and current_step == 1:  
        response = "Step 2 of 7: In another bowl, beat soft  
            butter, brown sugar, and vanilla sugar until  
            fluffy and pale."  
  
    # ...other logic for other userText goes here...  
  
    return response
```

END  
HERE



Step 2 of 7: In another bowl, beat soft butter,  
brown sugar, and vanilla sugar until fluffy and  
pale.

# 3. Text to Speech (TTS) Play the response out of Alexa's speaker.



# 1. Speech to Text (STT) aka Automatic Speech Recognition (ASR)

Unfortunately, we don't have access to this module, Amazon just sends us the output...

START HERE

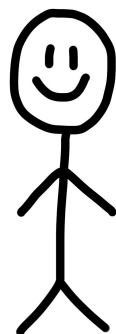


next

# 2. Run the code, and figure out what to say!

```
def decide_what_to_say(user_text, current_step):  
    if user_text == "next" and current_step == 1:  
        response = "Step 2 of 7: In another bowl, beat soft butter, brown sugar, and vanilla sugar until fluffy and pale."  
  
    # ...other logic for other userText goes here...  
  
    return response
```

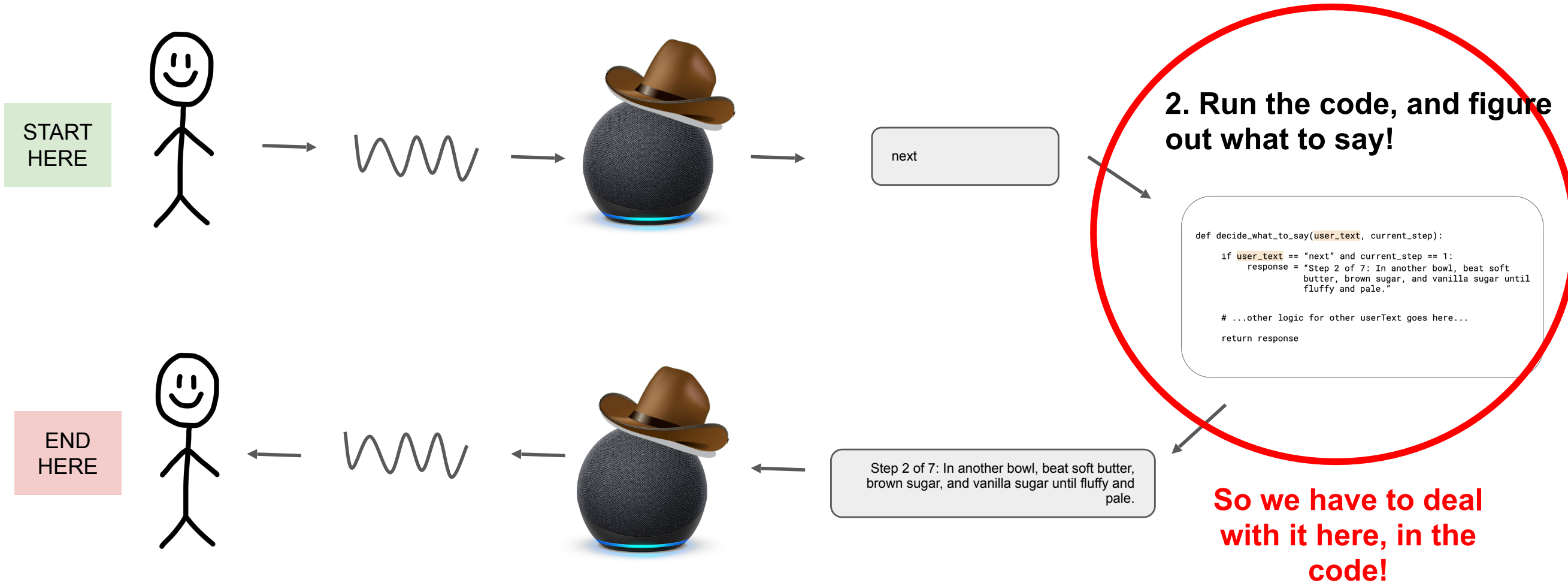
END HERE



Step 2 of 7: In another bowl, beat soft butter, brown sugar, and vanilla sugar until fluffy and pale.

# 3. Text to Speech (TTS) Play the response out of Alexa's speaker.

# 1. Speech to Text (STT) *aka Automatic Speech Recognition (ASR)*



# 3. Text to Speech (TTS) Play the response out of Alexa's speaker.

# Any ideas?

Now that we know what the problem is, how do we deal with it?

Ideas?

# Here's one way...

```
def decide_what_to_say(user_text, current_step):  
  
    next_list = ["next",  
                "desk",  
                "best",  
                "let's",  
                "list"]  
  
    for item in next_list:  
        if user_text == item and current_step == 1:  
            "Step 2 of 7: In another bowl, beat soft butter, brown sugar,  
            and vanilla sugar until fluffy and pale."  
  
    # ...  
  
    return response
```

What are some advantages and disadvantages to this method?

## Here's another way...

**Soundex** is a popular algorithm for dealing with these issues.

- Soundex takes words and turns them into codes.
- If multiple words have the same code, then they all **sound the same!**

One step in Soundex is swapping each of letters for numbers:

a, e, i, o, u, y, h, w	→ 0
b, f, p, v	→ 1
x, g, j, k, q, s, x, z,	→ 2
d, t	→ 3
l	→ 4
m, n	→ 5
r	→ 6

Why do you think the letters are grouped together in these specific groups?

Let's try it! What is the **Soundex code** for each of these **words**?

- "Robert"
- "Rupert"
- "Rubin"

What are some advantages and disadvantages to this method?

1. Keep the first letter of the word.
2. For the rest of the letters in the word:
  - a. Replace each letter with its corresponding number:

a, e, i, o, u, y, h, w → 0  
b, f, p, v → 1  
c, g, j, k, q, s, x, z → 2  
d, t → 3  
l → 4  
m, n → 5  
r → 6

- a. Replace all adjacent repeating numbers with just one number.
  - b. Remove all zeroes.
1. Now look at that first letter again. If its number (see chart) is the same as the number next to it, get rid of the number next to it!
2. How long is it now?
  - a. If it's less than length 4, pad it with zeroes until it's length 4.
  - b. If it's longer than length 4, remove from the end to make it length 4.
  - c. If it's length 4, awesome! Leave it!

# Next Steps

At one point, we asked clarification questions: For “star cooking” our taskbot may ask “Did you mean “start cooking”?”

But users get annoyed, so we try to avoid this as much as possible

Next Steps:

More principled machine learning approach, but we need data

We are hand labeling examples from our conversations to train a model



Challenge #2:  
Searching for the Right Task



Help me clean my bathroom.

**wikiHow** to do anything...  **PRO**   EDIT

 wikiHow is where trusted research and expert knowledge come together. Learn why people **trust wikiHow**

HOUSEKEEPING » CLEANING

# How to Clean Your Bathroom Fast

Co-authored by **Fabricio Ferraz**   
Last Updated: October 21, 2021  References

We all try to keep our homes as clean as possible, but some rooms can be a challenge. The bathroom is definitely one of the hardest rooms to keep because it gets used so often and is prone to plenty of messes, such as soap scum or mildew. If you have trouble keeping your bathroom clean and you have guests coming over shortly, you need a quick plan of attack to get it sparkling again. Focus on a few key areas of the space that will make a big impact -- and try to develop a more consistent cleaning schedule going forward.

 **Download Article**

**METHODS**

- 1 Cleaning on the Spot**
- 2 Starting Quick Cleaning Habits**
- 3 Keeping Your Bathroom Clean**

**OTHER SECTIONS**

-  **Questions & Answers**
-  **Video**
-  **Tips and Warnings**
-  **Related Articles**
-  **References**

# Step 1: Gather up any trash

If your trash can is full or the vanity is littered with crumpled tissues, your bathroom is definitely going to look messy. Getting rid of the garbage can instantly make the space appear cleaner. Start by emptying the trash can into a plastic garbage bag, and toss in any other trash that might be in the room. It's a good idea to leave the garbage bag hanging on the

...



# Default search engine is Elasticsearch

Only supports keyword, phrase or more-like-this query

WikiHow articles are indexed in Elasticsearch

(we don't have access to full WikiHow data)

Not very good

But our entire taskbot relies on the articles returned by the search engine!

Here's an example from Elasticsearch (results from WikiHow):  
***How to clean my restroom***

How to Overcome Public Restroom Embarrassment

How to use a public restroom

How to avoid germs in public restrooms

How to setup a restroom trailer

How to clean bathroom tile

And another example:

## ***How to draw on a wall?***

How to draw a brick wall

How to draw wall-e

How to get a toddler to stop drawing on walls

How to paint a concrete wall

How to hang pictures on a concrete wall

# Okay, so what would we like?

Move beyond just strict keyword match:

How to clean my **restroom**? → How to use a public **restroom**

Incorporate contextual meaning instead of keywords

Support broader topic, meaning or synonyms

Handle typos and variants

Be more “semantically” aware

# Our Approach

- Index vector representations of each title and description
  - Use machine learning-based model to encode text to a contextual word embedding that embeds the meaning of words
    - Sentence-BERT<sup>1</sup>
- Query and retrieve relevant documents by computing vector similarity search
- Use FAISS<sup>2</sup> to make it scalable and fast
  - RAM based index
  - C++ library (Python wrapper)
  - Supports add/remove/filter documents

<sup>1</sup>N. Reimers, I. Gurevych, [Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks](#) (2019), Proceedings of the 2019 Conference on Empirical Methods in NLP

<sup>2</sup>Johnson, Jeff, Matthijs Douze, and Hervé Jégou. "Billion-scale similarity search with gpus." *IEEE Transactions on Big Data* 7.3 (2019): 535-547.



# BERT? Embeddings?

Instead of representing a word by a sparse one-hot vector:

bathroom = [0 0 0 0 1 0 0 0 ... .. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Learn a dense “embedding”:

bathroom = [0.21 -0.91 -0.34 0.19 0.14 0.26 ... ]

The hope: words related to bathroom are nearby in this dense vector space!

BERT (and its variants) are a special way to learn smart embeddings over billions of documents (**pre-trained language models**)

# BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin   Ming-Wei Chang   Kenton Lee   Kristina Toutanova

Google AI Language

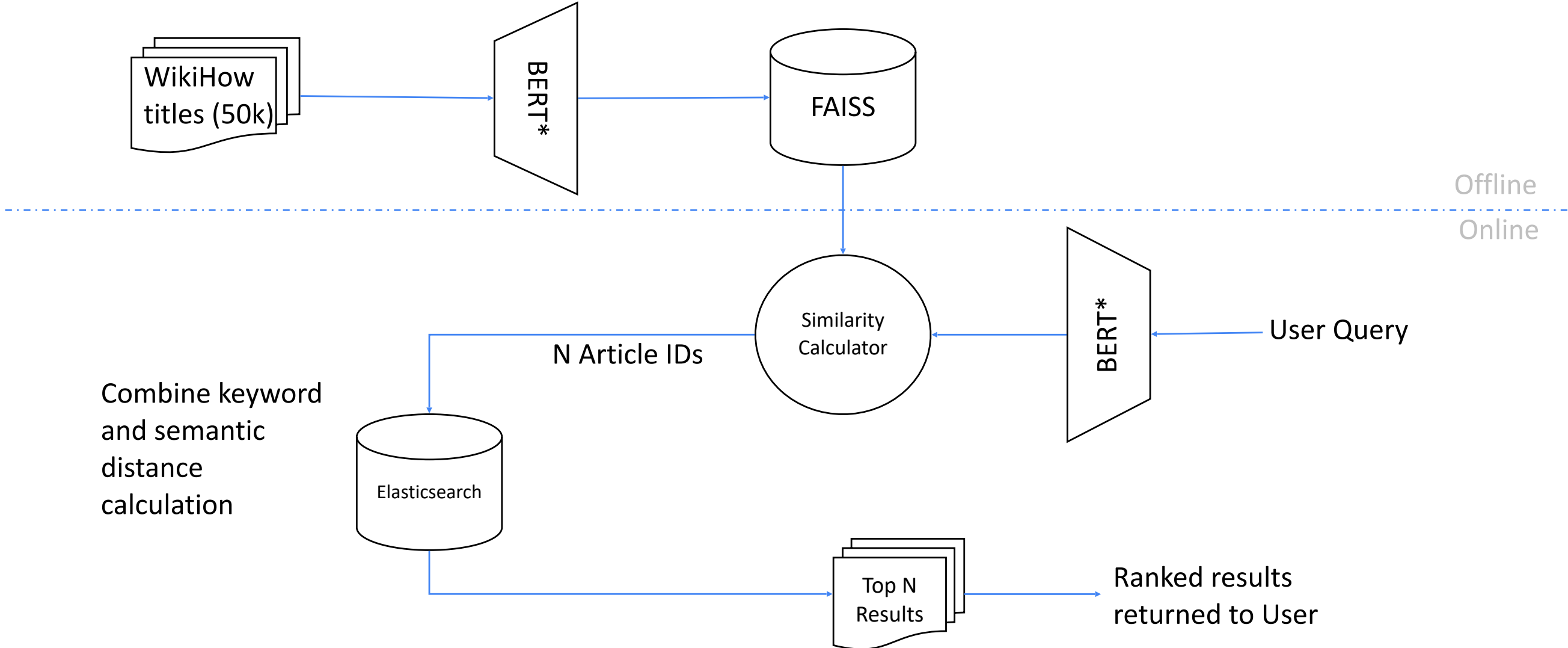
{jacobdevlin, mingweichang, kentonl, kristout}@google.com

## Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

# Semantic Search



# Example: How to clean my restroom

## Elasticsearch

- How to Overcome Public Restroom Embarrassment
- How to use a public restroom
- How to avoid germs in public restrooms
- How to setup a restroom trailer
- How to clean bathroom tile

## Our Approach

- How to clean a bathroom
- How to clean bathroom grout
- How to clean a bathroom sink
- How to clean a bathroom sink drain
- How to clean a jetted tub

# Example: How to color my wall

## Elasticsearch

- How to pick a color for an accent wall
- How to choose paint color for an bedroom
- How to paint a concrete wall
- How to change your eye color
- How to get colored contacts to change your eye color

## Our Approach

- How to paint a wall
- How to paint an interior wall
- How to paint walls near a ceiling
- How to paint designs on walls
- How to paint textured walls

# Example: How to draw on a wall

## Elasticsearch

- How to draw a brick wall
- How to draw wall-e
- How to get a toddler to stop drawing on walls
- How to paint a concrete wall
- How to hang pictures on a concrete wall

## Our Approach

- How to paint designs on wall
- How to paint a wall
- How to arrange pictures on a wall
- How to draw graffiti
- How to frame a wall

# Challenge #3: Determining User Intent

# What are Intents?

An intent represents a **task** or **action** the user wants to perform. It is a purpose or goal expressed in a user's utterance.

Wait, what's an utterance? Whatever it is that the ASR gives us (what the user says)



# Examples of intents

MoreIntent → tell me more about a particular step

NextIntent → give me the next step in a task

PreviousIntent → give me the previous step in a task

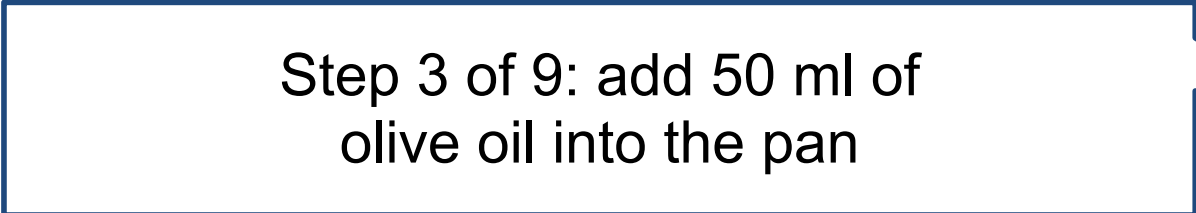
StartIntent → start a particular task

SetTimerIntent → set a timer (like for baking)

NewSearchIntent → start a new search

...

# Which Intent?



Step 3 of 9: add 50 ml of  
olive oil into the pan



Yes.

Which intent does this map to?  
Or, what does the user expect?

# Which Intent?

Step 3 of 9: add 50 ml of  
olive oil into the pan



Yes.

Which intent does this map to?  
Or, what does the user expect?

Step 4 of 9: heat the oil for  
5 minutes with medium heat



# Which Intent?

Step 3 of 9: add 50 ml of  
olive oil into the pan



I don't have olive oil



# Which Intent?

Step 3 of 9: add 50 ml of olive oil into the pan



I don't have olive oil



You can substitute butter for the olive oil if you prefer.



# Which Intent?

Step 3 of 9: add 50 ml of  
olive oil into the pan



I don't have olive oil



# Which Intent?

Step 3 of 9: add 50 ml of  
olive oil into the pan



I don't have olive oil



Do you want to add olive oil  
into your shopping list?



# Which Intent?

Step 3 of 9: add 50 ml of  
olive oil into the pan



I don't have olive oil



Do you want to add olive oil  
into your shopping list?



Yes.





# Which Intent?

Step 3 of 9: add 50 ml of olive oil into the pan



I don't have olive oil

Do you want to add olive oil into your shopping list?



Yes.

Great, olive oil has been added to your shopping list



# Determining User Intent

Given a user utterance (and maybe the history of the conversation), our goal is to map the utterance to an intent

Ideas?

Our Approach: BERT! (again)

# Challenge #4: Answering Questions

# Users Ask Questions (a lot!)

We handle three kinds of question answering models:

- Factoid QA Model (EVI from Amazon): answer factoid questions
- Extractive QA Model: answer task or recipe related questions, e.g. temperature, cooking time, recipe rating, ingredient substitute, etc.
- Abstractive QA Model: answer task related questions

# Factoid QA

Respond to factoid questions.



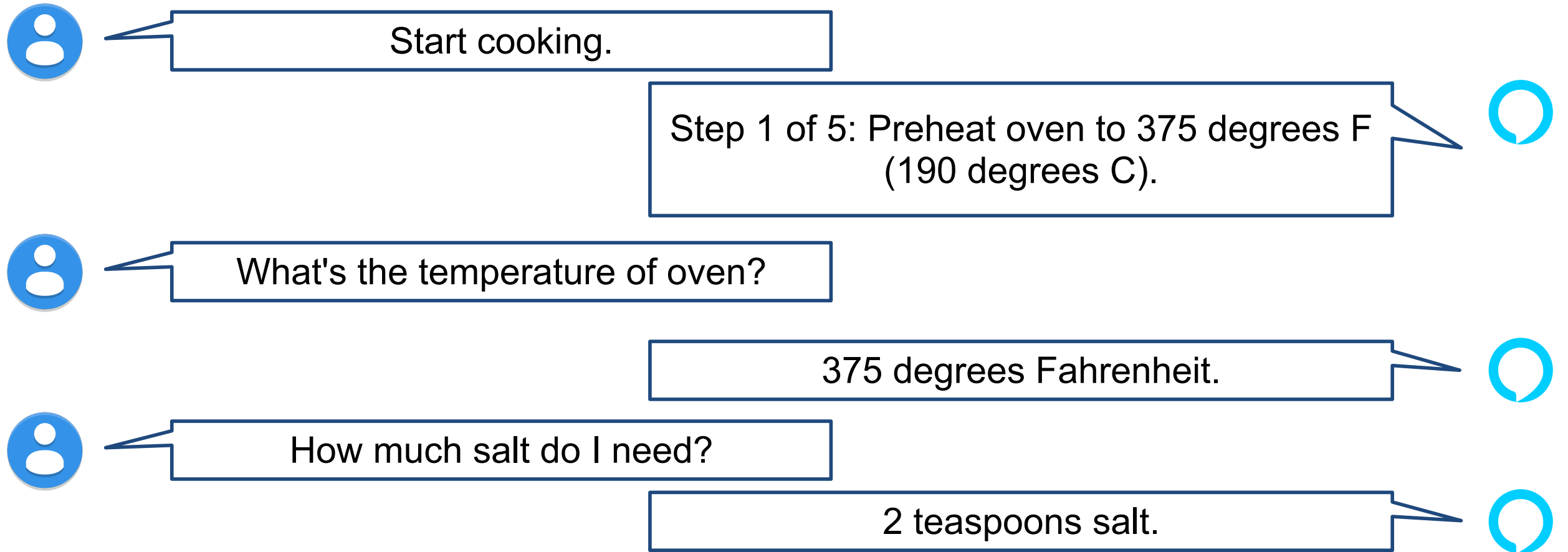
How many planets of the galaxy?

The Milky Way has 10,000,000,000 planets.



# Extractive QA

Extract answers from **data** or knowledge base.



# Extractive QA

Extract answers from data or **knowledge base**.



I don't have lemonade

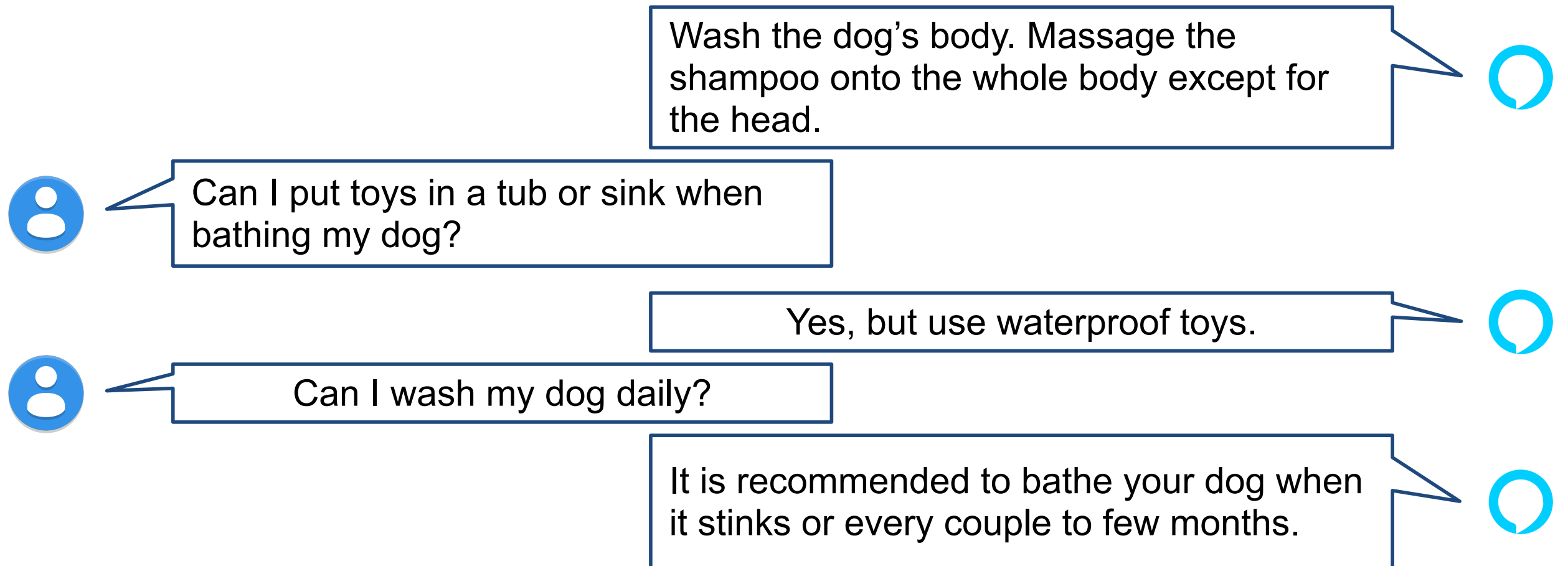
You may substitute Bottled lemon or lime juice or white vinegar for lemon or lime juice (fresh).





# Abstractive QA

Generate responses from generative model.



Challenge #5:  
Keeping the User Engaged

# Empathic Responses

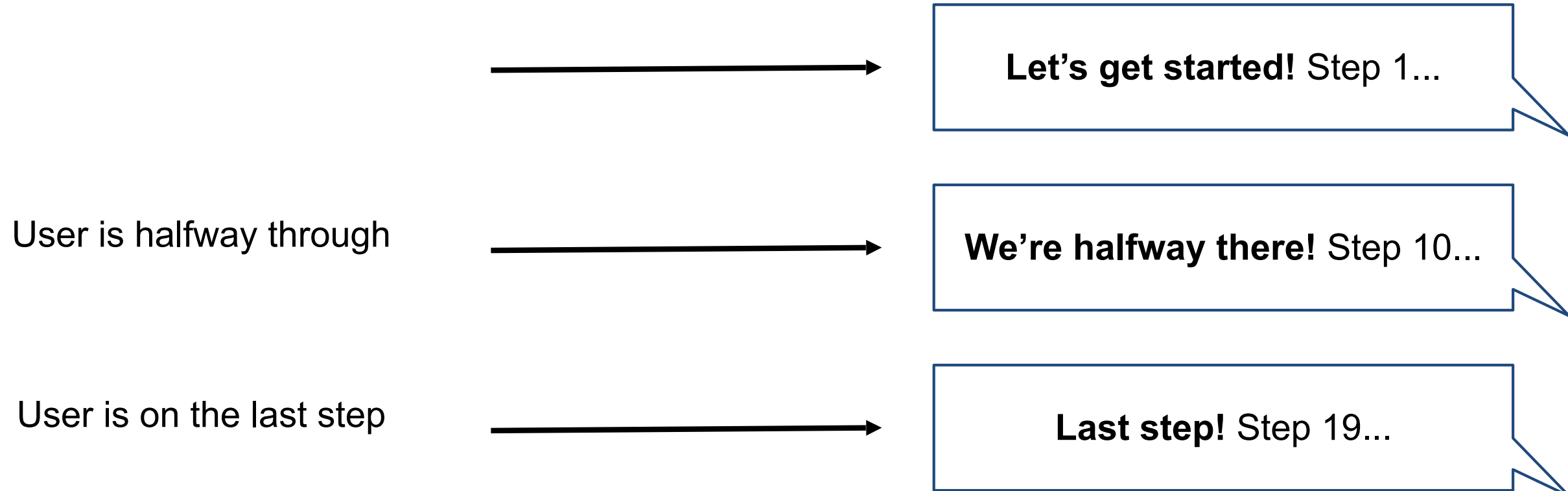
## Problem:

How can we design for the **emotional aspects** of **encouraging users to complete tasks?**

## Current:

- Rule-based Approach
  - *Example 1: Conversationalize Steps*
  - *Example 2: User is still working*
  - *Example 3: User tells us their name*
- “Bad Conversation” Model

# Rule-Based Example 1



*Beginnings of **conversationalizing** the steps...*

# Rule-Based Example 2

**User is is still working and needs more time**

*"I'm still working..."*

*"I am still mixing the eggs..."*



O.K., no rush, take your time! I'm here to help whenever you're ready.

# Rule-Based Example 3

**User tells us their name**

*“I’m Reveille...”*

*“My name is Reveille...”*

*“I am called Reveille...”*



Hi! I’m so happy you’re here,  
Reveille.

# “Bad Conversation” Model

“Bad Conversation”  
Model categorizes current  
conversation as “bad”



**It seems like this conversation isn't going well.** I'm sorry about that, I'm still a work in progress... If you want to give me another chance, you can try telling me the same thing in a different way, and I'll do my best to understand. If not, I'm always learning, so please come back to chat again soon! To exit taskbot, say: stop.

# Future Directions

1. Classify and respond to **more situations** using rules and the **intent classification model**:
  - User wanting to chitchat
    - *“Let’s talk”*
    - *“How old are you”*
  - User who needs more guidance
    - *“i don’t know what you’re talking about”*



# Summary

Conversational AI is an exciting area!

Lots of open challenges ...

- Both in terms of engineering

- And in terms of fundamental research advances

Preparation?

- Strong understanding of algorithms and data structures

- Probability, statistics, linear algebra

- Machine learning, AI, NLP, Information Retrieval, ...