CHRIS GERPHEIDE <CHRIS@BE-SPOKE.IO> 2019-07-22

# CREATING A CHATBOT FROM SCRATCH

BESPOKE

Bebot / Chatbot architecture and programming

# Implement the guts of a chatbot!

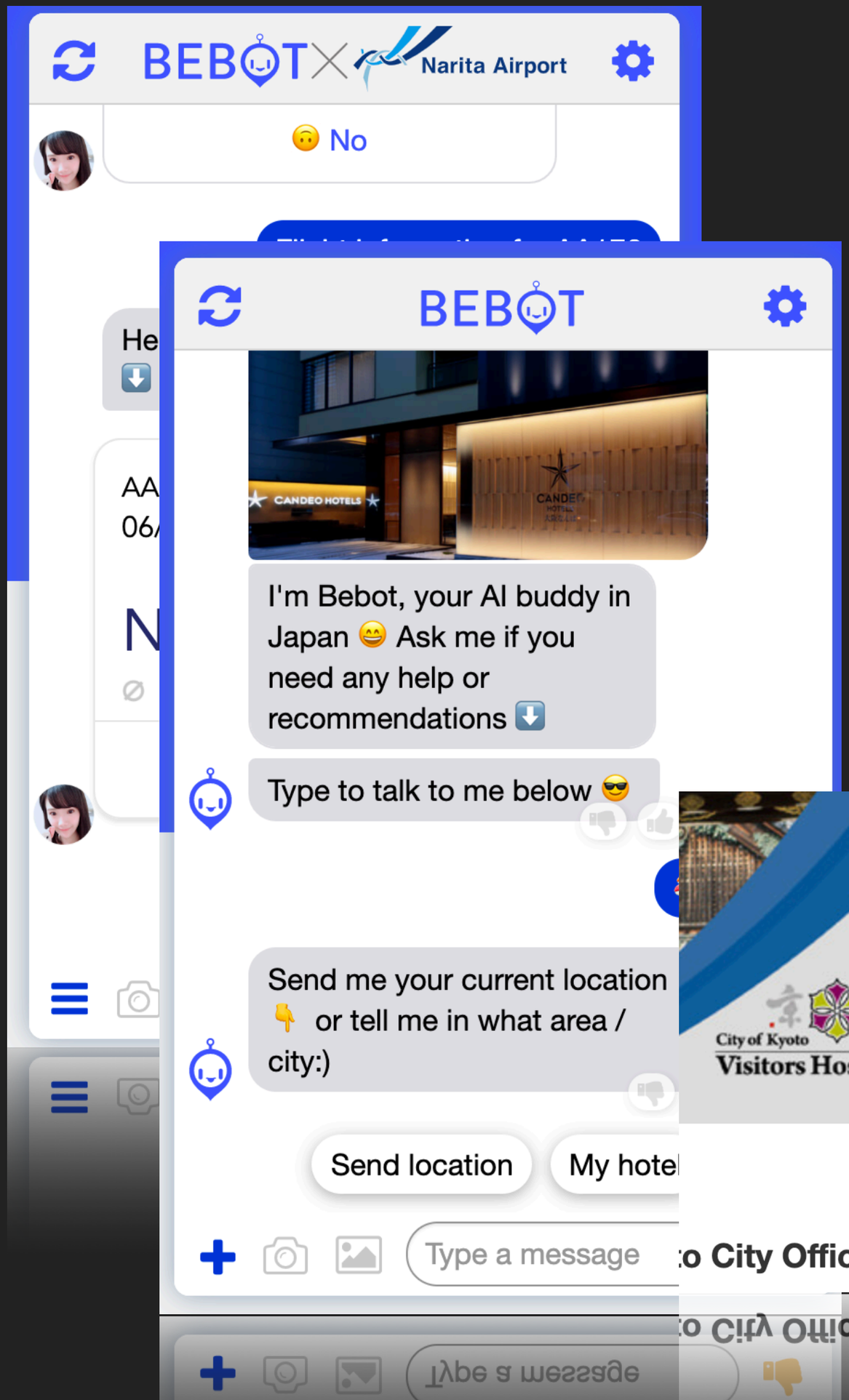
**With python and scikit-learn**

Deploy

BESPOKE

# REBOT

BESPOKE



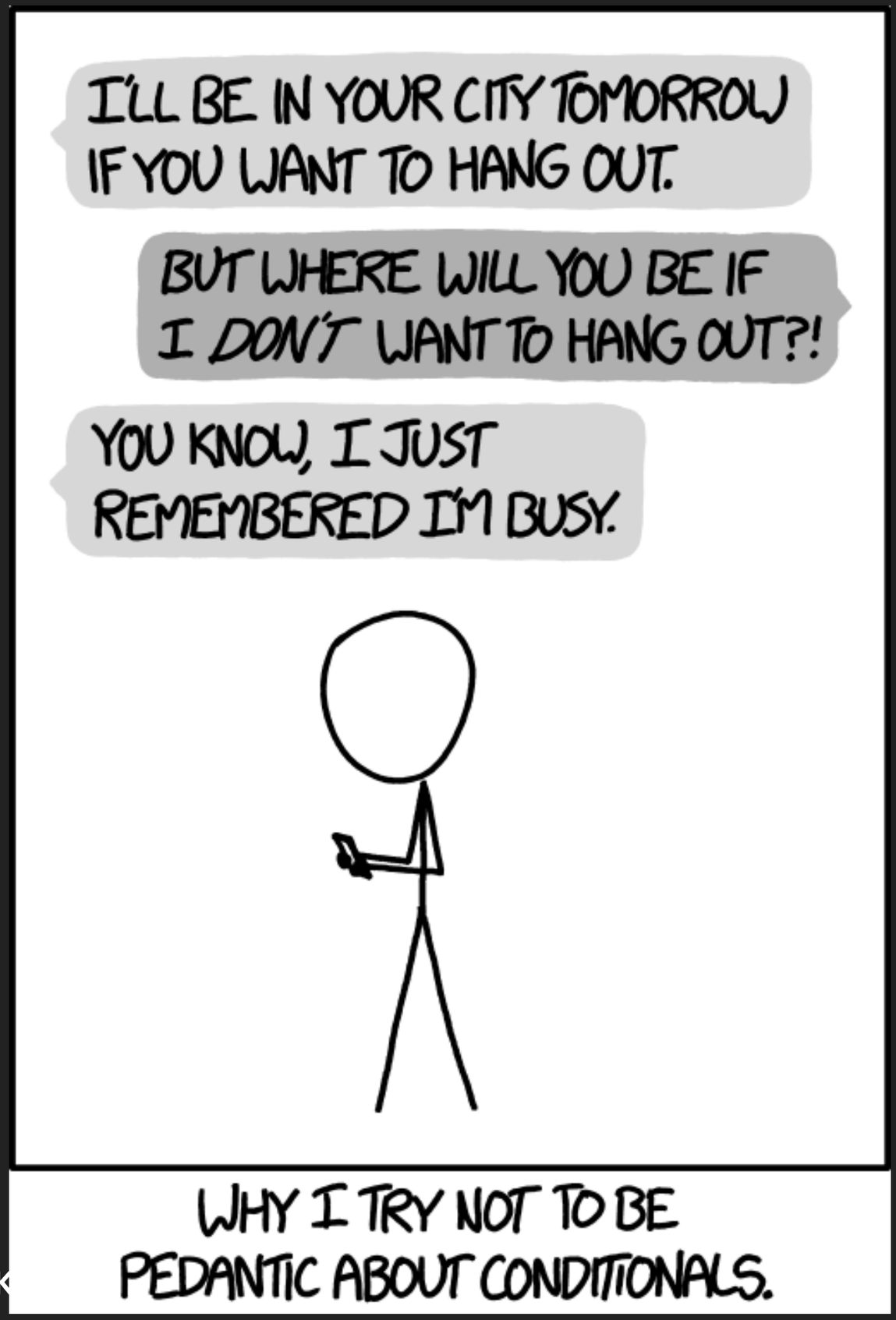

- AI-powered chatbot
- Custom conversations, live support, and insights
- Clients include Narita airport, Tokyo station, hotels, Japanese Gov't for disaster relief
- ~30,000 users daily

# CHATBOT ARCHITECTURE

BESPOKE

# OPTION 1: HUMANS

# OPTION 2: RULE-BASED




xkcd.com

# OPTION 3: PREDICTIVE — RETRIEVAL-BASED

## IF MACHINE LEARNING IS THE ANSWER, THEN WHAT IS THE QUESTION?

Shoham, Powers, Grenager 2006

# TRAINING

"CLASS"

"LABEL"

"INTENT"

The topic or question
the user is asking about

"MODEL"

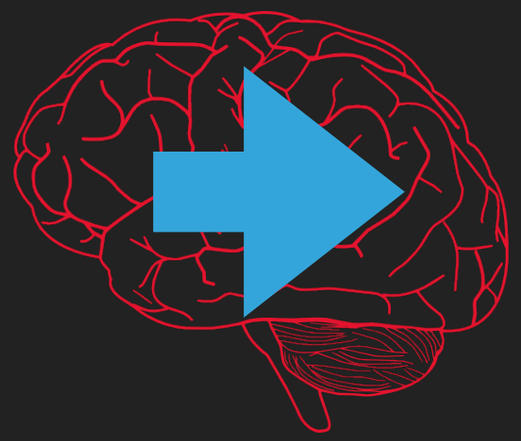
"CLASSIFIER"

**where_is_my_hotel** : [
'Where is my hotel?',
'Hotel location?',
'How do I get to the hotel?'
]

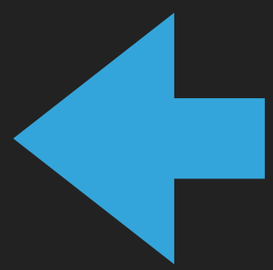
**when_is_checkout_time** : [
'When is checkout time?',
'When do I need to check out?'
]

# PREDICTION AND RETRIEVAL

“ohi how get 2 hotel kthx ¯_(ツ)_/¯”



where_is_my_hotel



“Your hotel is located across from Shibuya Station. Check out these directions:

https://goo.gl/aoeu”

# PREDICTION, ONE LEVEL DEEPER…

where_is_my_hotel : [
‘Where is my hotel?’,
‘Hotel location?’,
‘How do I get to the hotel?’
]

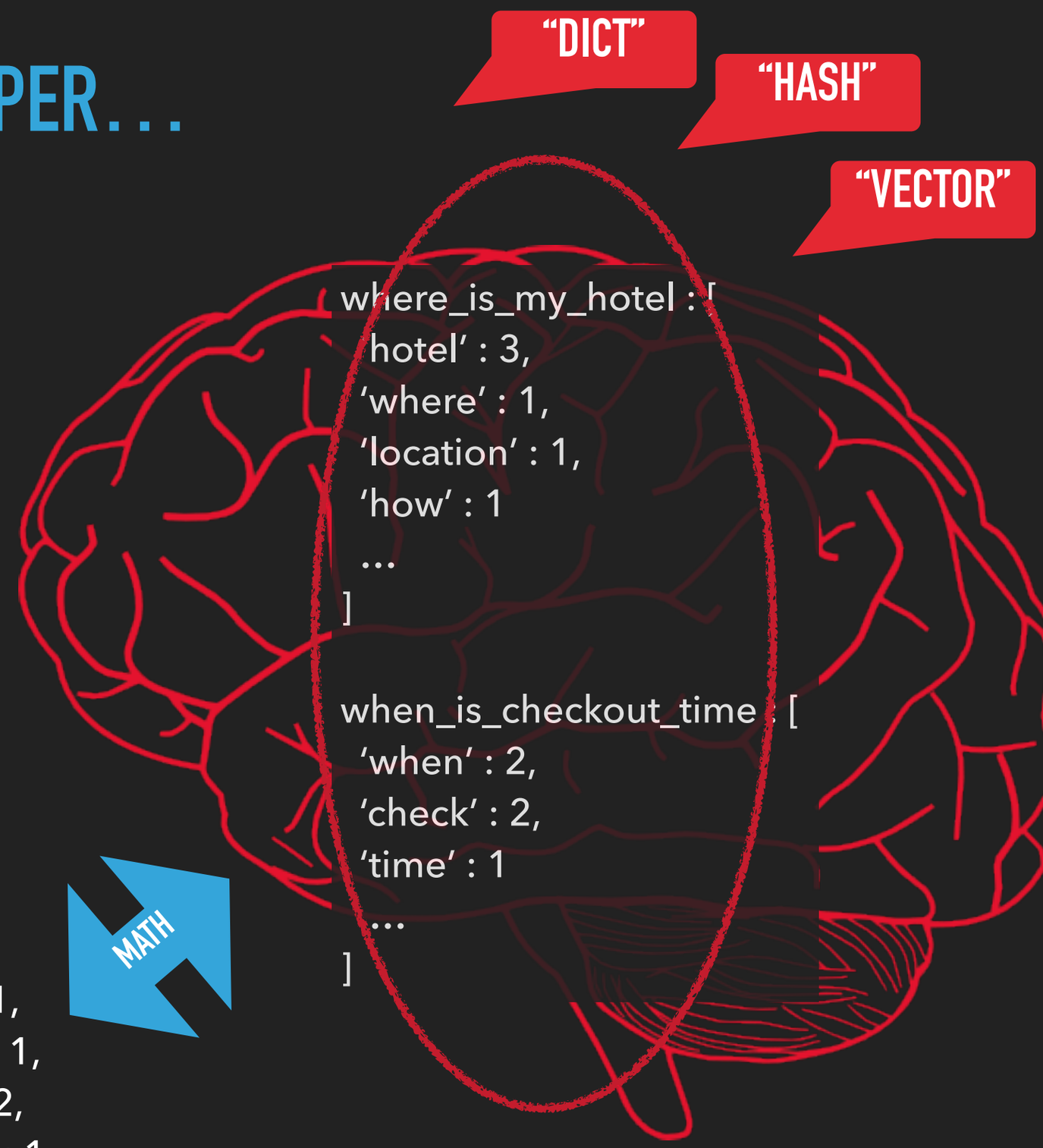
when_is_checkout_time : [
‘When is check-out time?’,
‘When do I need to check out?’
]

“ohi how get 2 hotel kthx ¯_(ツ)_/¯”

[
‘ohi’ : 1,
‘how’ : 1,
‘get’ : 2,
‘hotel’ : 1
…
]

MATH

BESPOKE



# THE TEST SET

accuracy, but not very granular

| Query | Expected Label | Prediction Probabilities | Prediction Result | |
|---|---|---|---|---|
| "Where hotel plz" | where_is_my_hotel | where_is_my_hotel: 0.72<br>when_is_checkout_time: 0.26 | where_is_my_hotel | ✓ |
| "How can I go to the hotel?" | where_is_my_hotel | where_is_my_hotel: 0.95<br>when_is_checkout_time: 0.18 | where_is_my_hotel | ✓ |
| "When is checkout?" | when_is_checkout_time | where_is_my_hotel: 0.14<br>when_is_checkout_time: 0.78 | when_is_checkout_time | ✓ |
| "When do I go to the checkout location?" | when_is_checkout_time | where_is_my_hotel: 0.42<br>when_is_checkout_time: 0.37 | where_is_my_hotel | ✗ |

# PRECISION AND RECALL

**Precision**: Ratio of true positive predictions to the total *predicted* positives

2 / 3 = 0.66

**Recall**: Ratio of true positive predictions to the total *expected* positives

2 / 2 = 1.00

where_is_my_hotel

| | Expected | Predicted |
|---|---|---|
| "Where hotel plz" | POS | POS |
| "How can I go to the hotel?" | POS | POS |
| "When is checkout?" | NEG | NEG |
| "When do I go to the checkout location?" | NEG | POS |

"TRUE NEGATIVE"

"FALSE POSITIVE"

# PRECISION AND RECALL

CONTROLLING FALSE ALARMS

**Precision**: Ratio of true positive predictions to the total *predicted* positives

**Recall**: Ratio of true positive predictions to the total *expected* positives

*High **precision** and low **recall***

| | Expected | Predicted |
|---|---|---|
| "Where hotel plz" | POS | NEG |
| ... | POS | NEG |
| | POS | NEG |
| | POS | NEG |
| | POS | NEG |
| | POS | NEG |
| | POS | NEG |
| | POS | NEG |
| | POS | NEG |
| | POS | NEG |
| | POS | POS |

make_reservation
(a lot of work, low risk if missed)

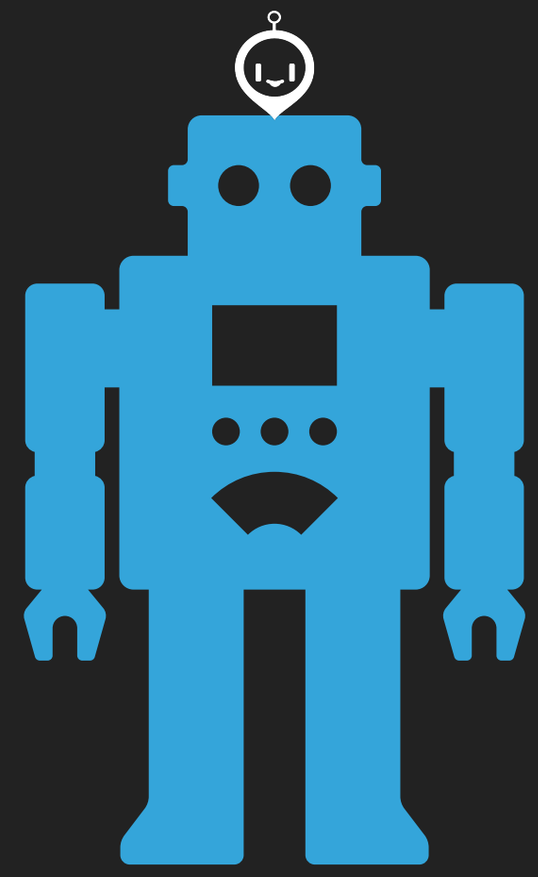
*Low **precision** and high **recall***

| | Expected | Predicted |
|---|---|---|
| "Where hotel plz" | NEG | POS |
| ... | NEG | POS |
| | NEG | POS |
| | NEG | POS |
| | NEG | POS |
| | NEG | POS |
| | NEG | POS |
| | NEG | POS |
| | NEG | POS |
| | NEG | POS |
| | POS | POS |

call_hospital
(high risk if missed)

(the guts of)

# CREATE A CHATBOT

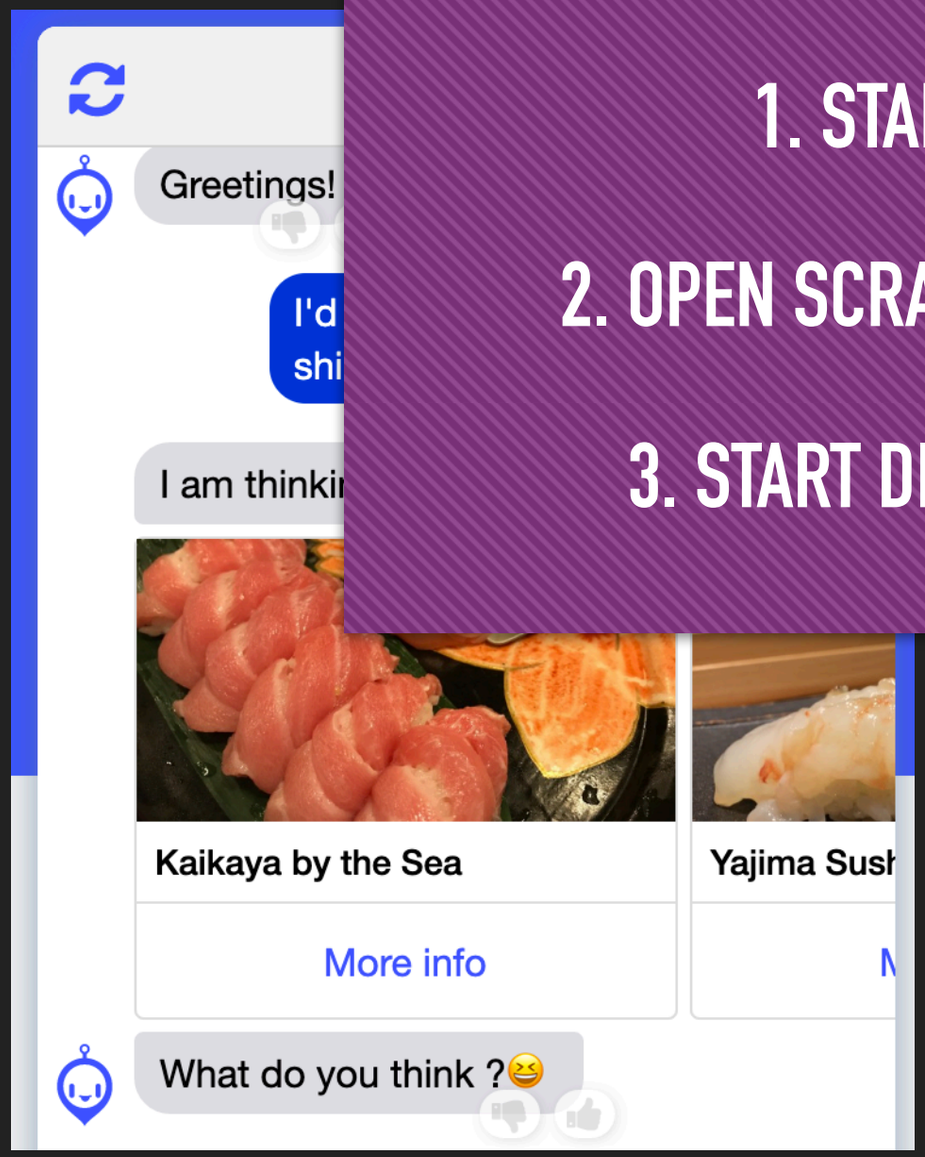


BESPOKE

# BE CREATIVE!

https://www.facebook.com/NatGeoGenius/

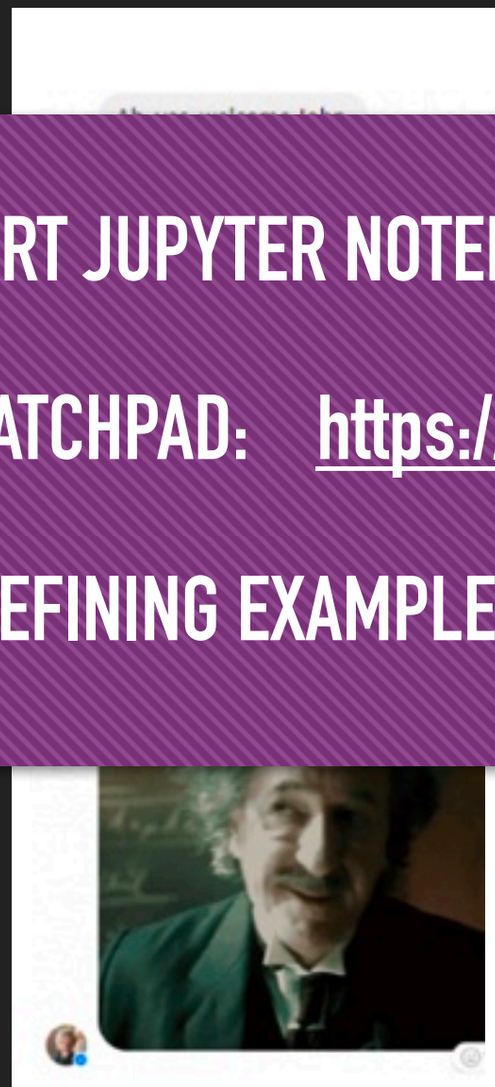
bebot.io




## 1. START JUPYTER NOTEBOOKS!

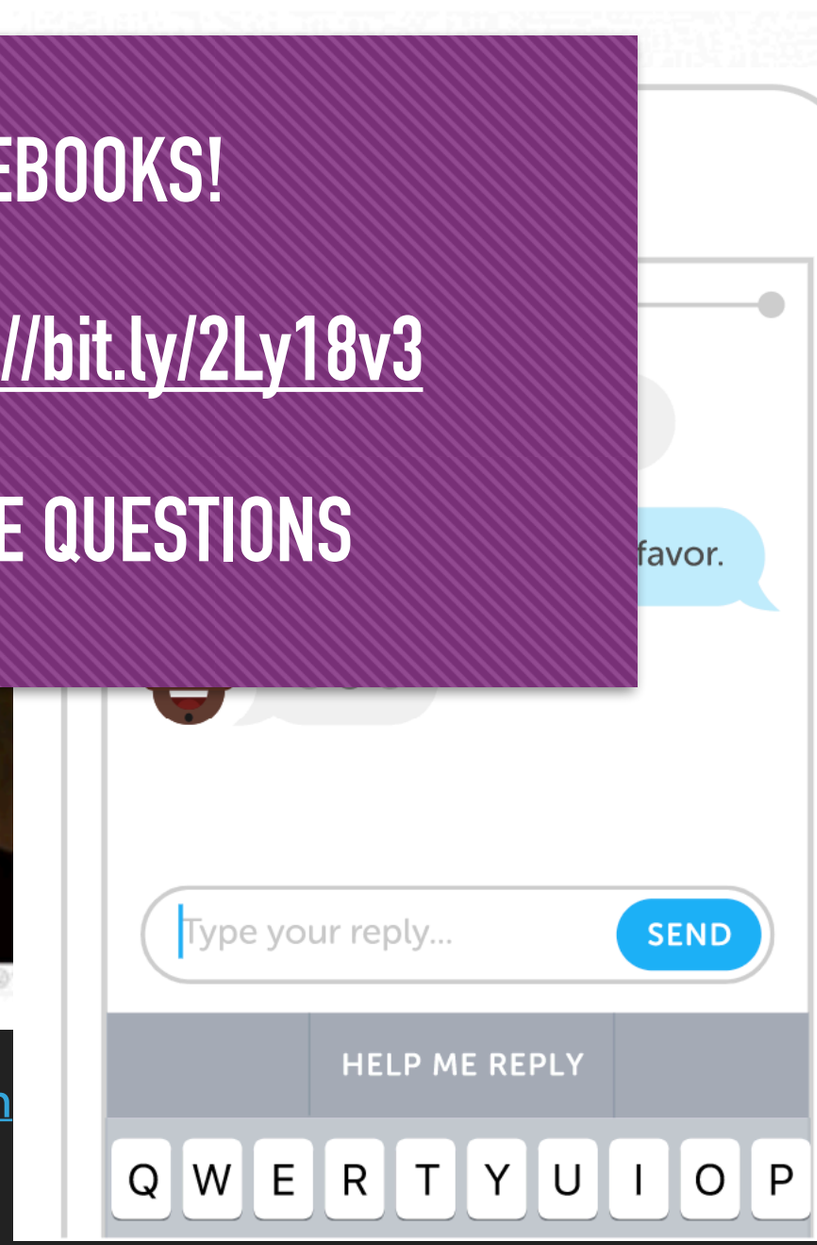
## 2. OPEN SCRATCHPAD: https://bit.ly/2Ly18v3

## 3. START DEFINING EXAMPLE QUESTIONS



bots.duolingo.com

BESPOKE



# WORKSHOP CHALLENGES

1. Return the answer
2. Exclude unimportant words (“stop words”)
3. Handle synonyms (e.g. “lobby” = “front desk”)
4. Handle typos
5. Return “unknown”
6. Handle a parameter (“set my check out time to 3pm”)

# CHALLENGES LEFT TO THE READER

- Other languages
- Conjugation and punctuation
- Domination of frequent words or intents
- Conversation state
- Conversation design

BEBOT

Amazon Lex

Dialogflow

# DEPLOY

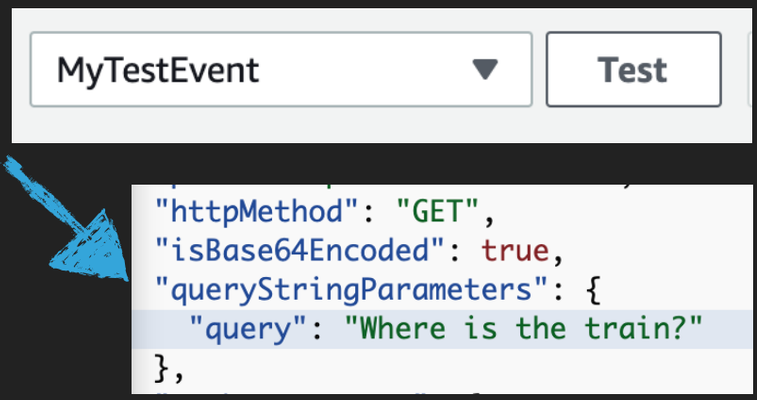
BESPOKE

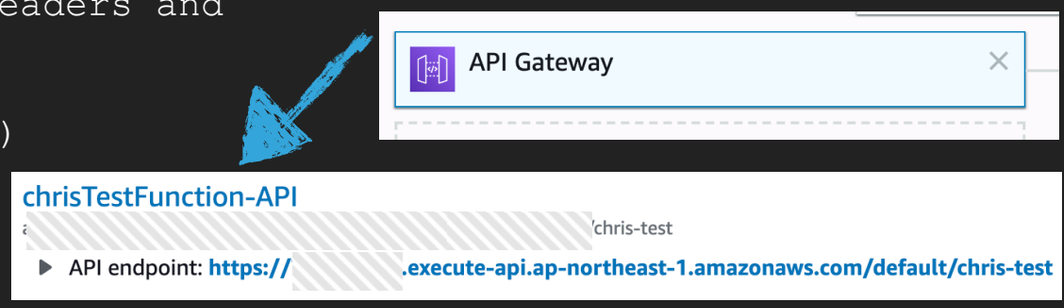
# AWS LAMBDA + API GATEWAY

Serverless App Repository:
Microservice-http-endpoint-python3

Create a deployment package with sklearn

```
import json

print('Loading function')

def respond(err, res=None):
    return {
        'statusCode': '400' if err else '200',
        'body': err.message if err else json.dumps(res),
        'headers': {
            'Content-Type': 'application/json',
        },
    }


def lambda_handler(event, context):
    '''Demonstrates a simple HTTP endpoint using API Gateway. You have full
    access to the request and response payload, including headers and
    status code.
    '''
    #print("Received event: " + json.dumps(event, indent=2))

    operation = event['httpMethod']
    if operation == 'GET':
        query = event['queryStringParameters']['query']
        # To start, you could simply train the model on every run
        # Then, run the predict method here.
        return respond(None, "Hi! Your question was: {}".format(query))
    else:
        return respond(ValueError('Unsupported method "{}"'.format(operation)))
```



chris@be-spoke.io

Chris Gerpheide

# THANKS!

# (AND WE'RE HIRING!)

Current openings:

- Chatbot R&D Team Lead
- Junior NLP engineer (upcoming)
- Full-stack developers (upcoming)

be-spoke.io/jobs

BESPOKE