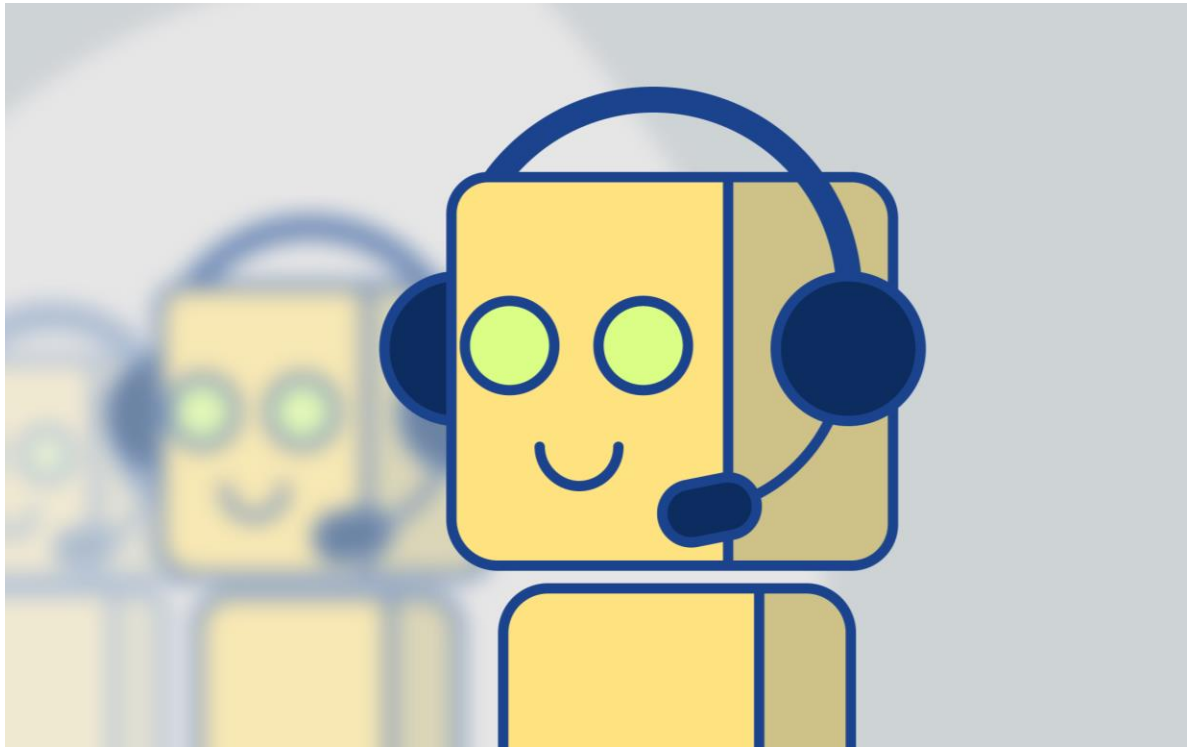


CHATBOTS

https://ravitha.github.io/

QUESTIONS

01

What are Chatbots

02

What they are useful for

03

How to create bots

04

What technology do they use

CHAT + BOT

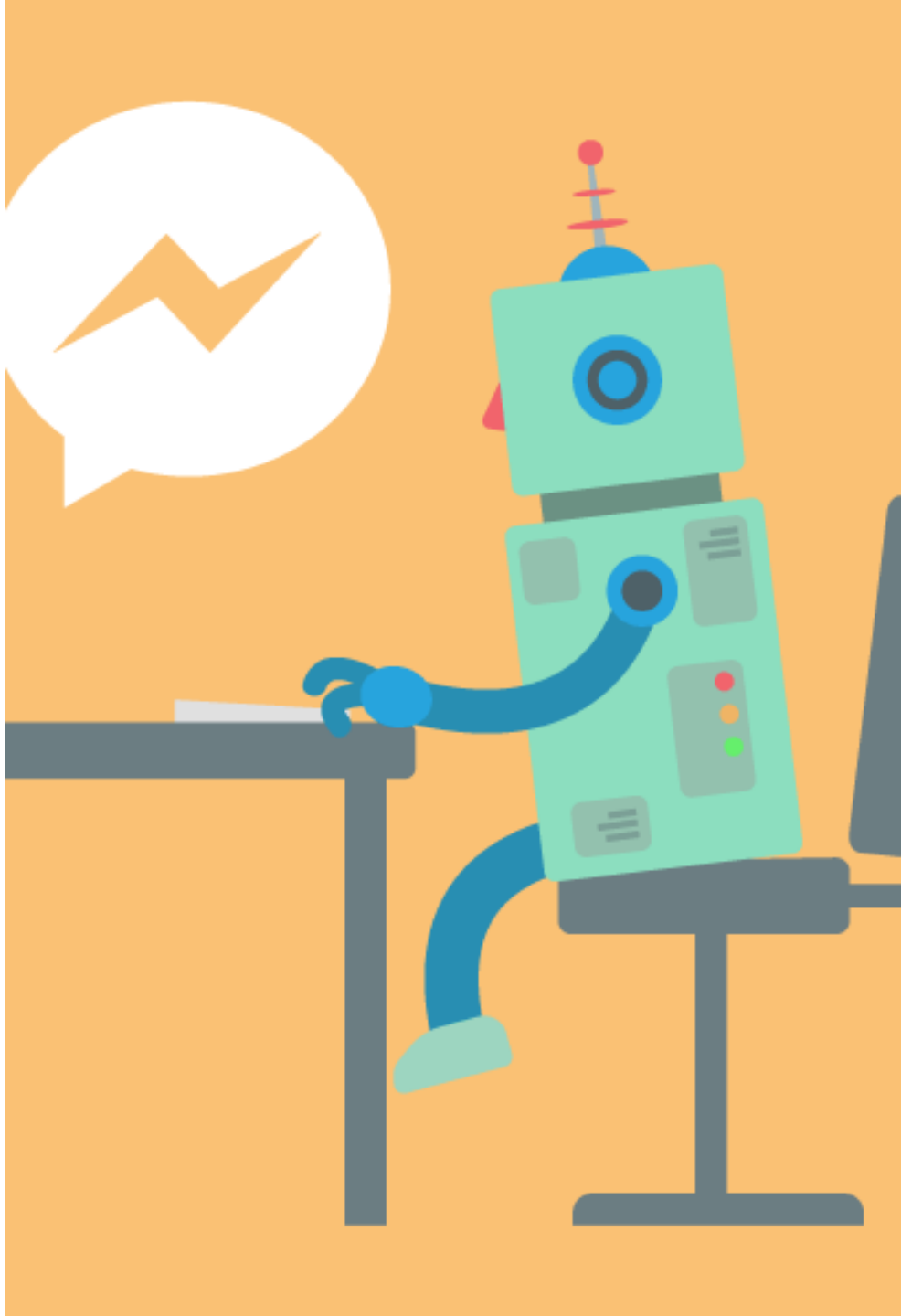
Realtime exchange of messages

+

ROBOT – Autonomous program on a network

→

A computer program designed to simulate conversation with human users, especially over the Internet.



CONVERSATIONAL INTERFACES

Device (or) software which enables the user to interact with the bot

establishing conversation

They are not mere text based interfaces

CONVERSATIONAL UI EXAMPLE

E.g. “I would like to reserve a cargo van for 1 pm this Friday”. This statement essentially combines a bunch of procedural steps together.

Q: “What would you like to do?”, A: “Make a reservation”

Q: “What would you like to reserve?”, A: “A cargo van”

Q: “What day would you like to reserve it for?“, A: “Friday”

Q: “At what time?“, A: “1 pm”

DESIGNING CONVERSATIONAL UI

Domain knowledge: What does a user expect this bot to understand?

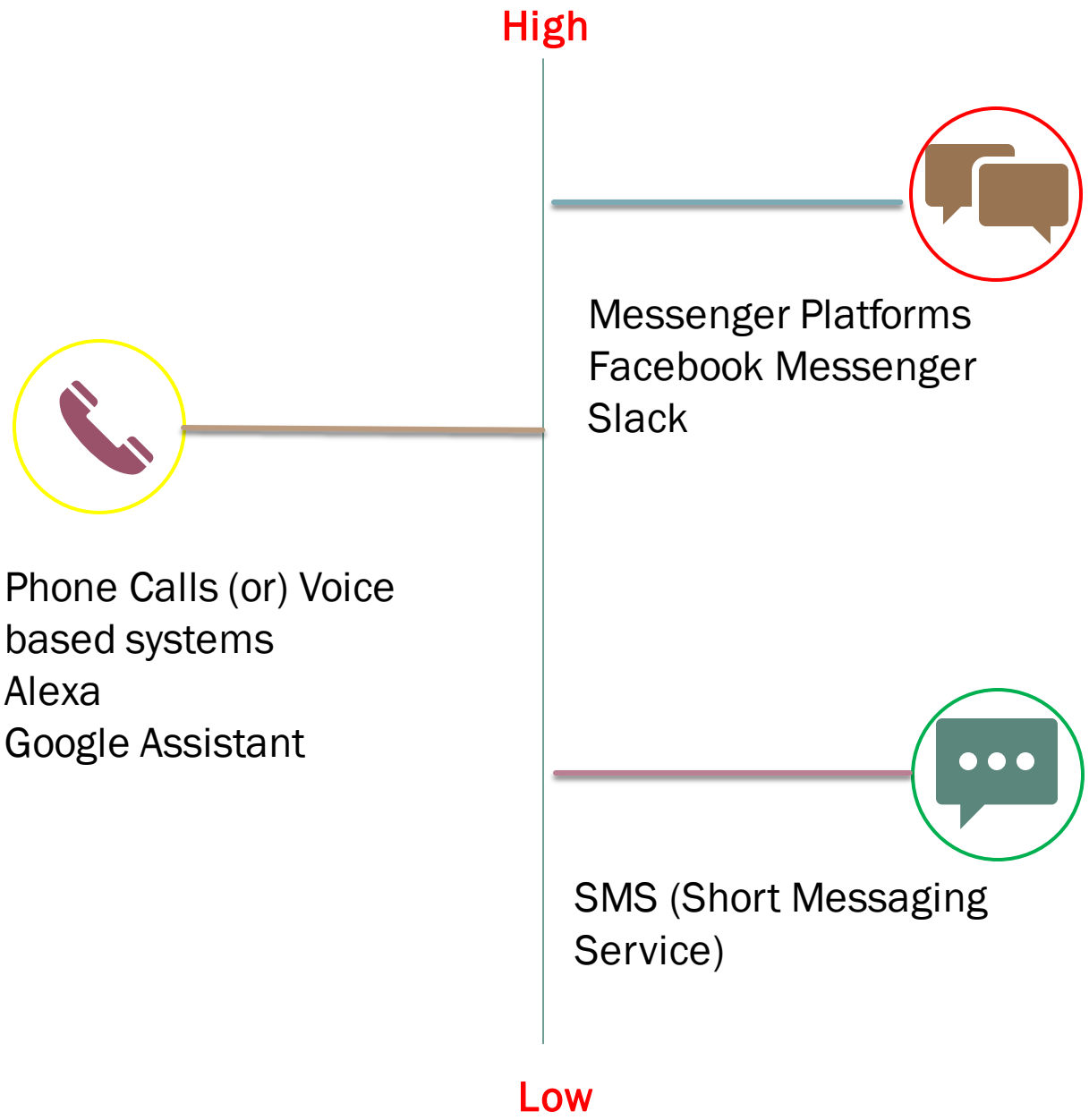
Personality: What tone or vocabulary does the bot employ?

ELIZA

Therapy Session

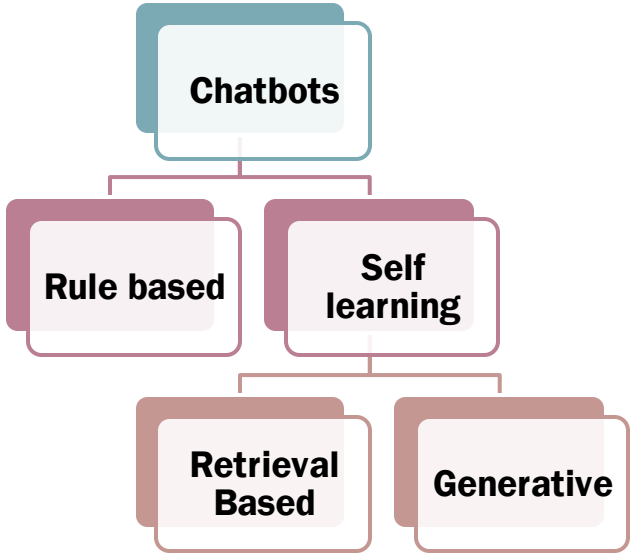
Rogerian Therapist

PLATFORMS

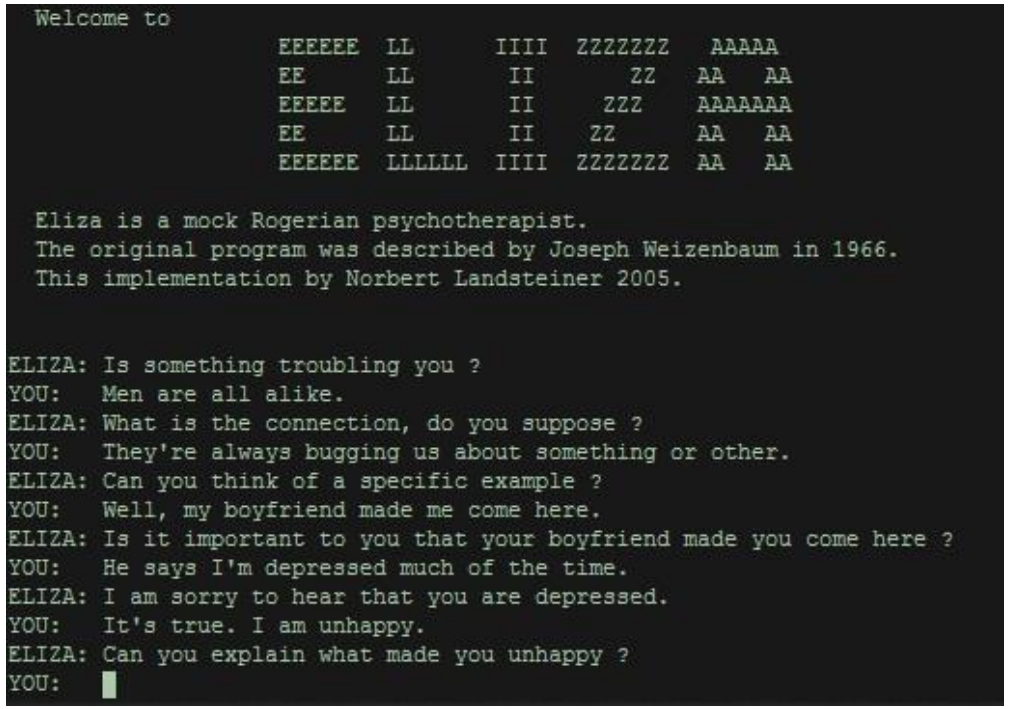

High
Messenger Platforms
Facebook Messenger
Slack
Phone Calls (or) Voice based systems
Alexa
Google Assistant
SMS (Short Messaging Service)
Low

EVOLUTION OF CONVERSATIONAL AI

Scripted Chatbots	Intent Recognizers	Virtual Agents	Human Like Advisor
• Uses key phrases in the request and provides a pre-defined response	• Machine Learning • Interpret the relationship between the words to understand the request	• Can understand the human request • Provide personalized answers • Connects to other systems to leverage the data	• It can understand and reason its environment like a human



PAST WORKS



Turing Test – 1950

ELIZA – 1966 (MAD-Slip Symmetric List Processor) (IBM 704)

Responses to messages are provided via text based terminal

Most famous implementation of ELIZA is **DOCTOR[1]**

PARRY – 1970

Jabberwacky – 1988

Dr.Sbaitso - MS-DOS based personal computers

ALICE - 1975

[1] http://psych.fullerton.edu/mbirnbaum/psych101/Eliza.htm

CURRENT TRENDS

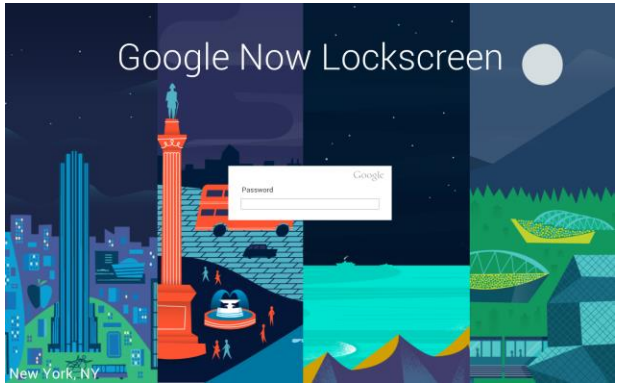
Apple Siri – 2011

Google Now – 2012

Microsoft Cortana - 2013

Amazon Alexa – 2014

Google Assistant



SCRIPTED BOTS

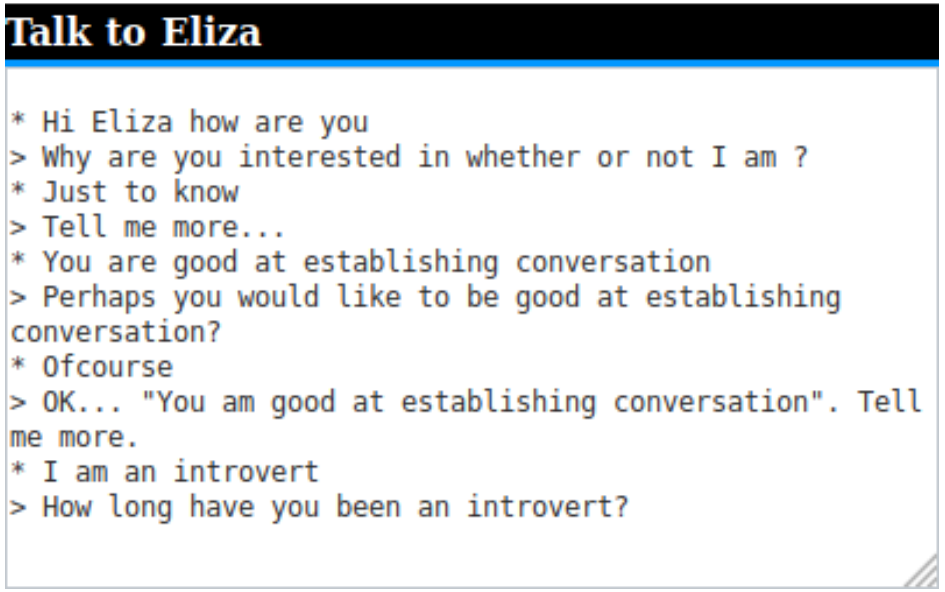
SIMPLE BOT

It searches the user's utterance for one or more known keywords and returns one of several possible responses.

```
GREETING_KEYWORDS = ("hello", "hi", "greetings", "sup", "what's up",)
GREETING_RESPONSES = ["sup bro", "hey", "*nods*",
"hey you get my snap?"]

def check_for_greeting(sentence):
        """If any of the words in the user's input was a greeting, return
        a greeting response"""
        for word in sentence.words:
                if word.lower() in GREETING_KEYWORDS:
                        return random. choice(GREETING_RESPONSES)
```

ELIZA



http://psych.fullerton.edu/mbirnbaum/psych101/Eliza.htm

ELIZA IMPLEMENTATION

Regular Expressions

- Match messages against known patterns
- Extract Key phrases
- Transform Sentences Grammatically (Conjugative verbs)

https://dhconnelly.com/paip-python/docs/paip/eliza.html

Rules

Pattern	Responses
If (.*)	Do you really think its likely that {0} What do you think about {0} Really- if{0}
Do you think (.*)	If{0}?Absolutely. No chance
I want (.*)	Why do you want {0} What's stopping you from getting {0} What would it mean if you got {0}
Do you remember (.*)	Did you think I would forget{0} Why haven't you been able to forget{0} What about {0} Yes.. And ?

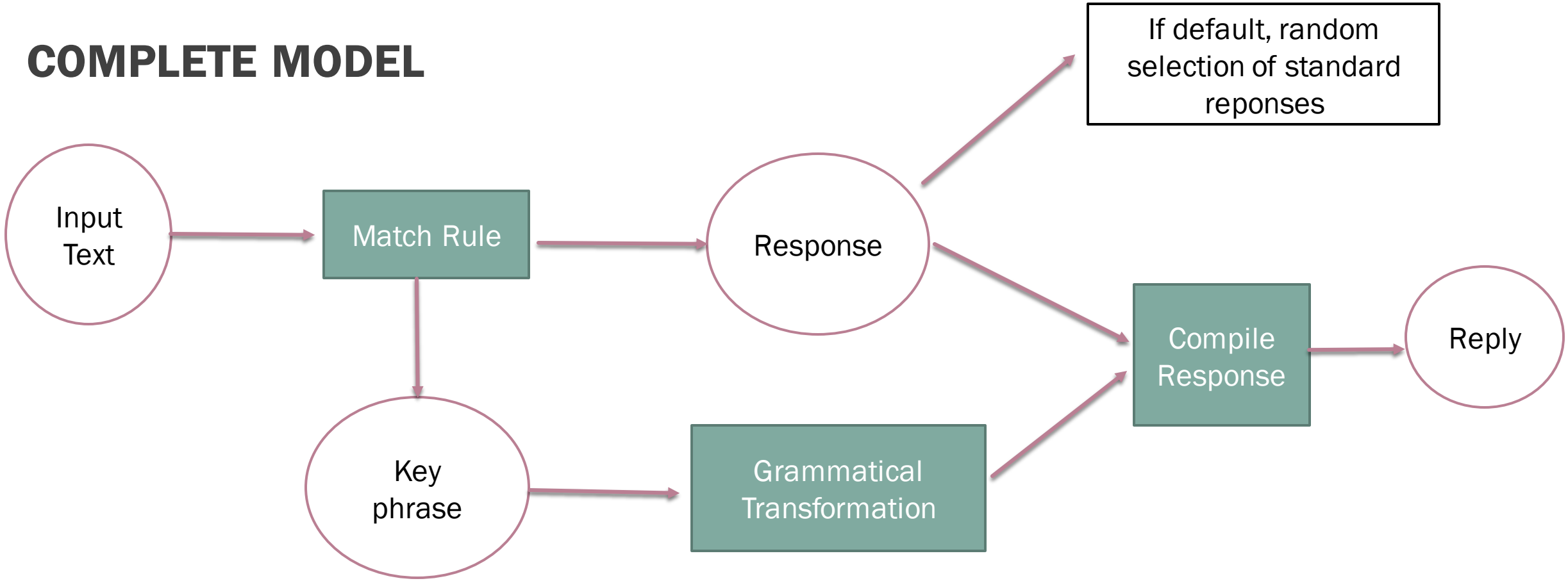
Match messages against known patterns

```
def match_rule(rules, message):
    response, phrase = "default", None
    # Iterate over the rules dictionary
    for pattern,responses in rules.items():
        # Create a match object
        match = re.search(pattern, message)
        if match is not None:
            response = random.choice(responses)
            # Choose a random response
            if '{0}' in response:
                phrase = match.group(1)
    # Return the response and phrase
    return response, phrase
```

Grammatical Transformation

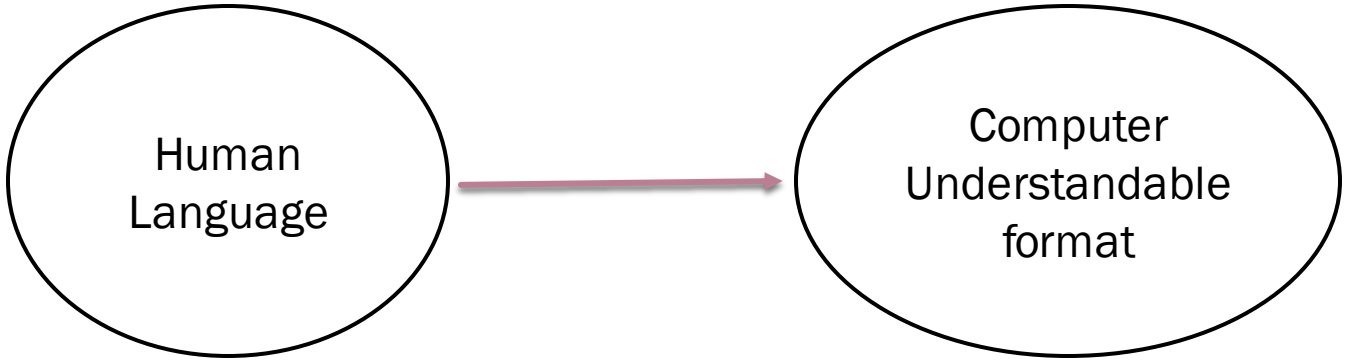
```
def replace_pronouns(message):
  message = message.lower()
  if 'me' in message:
    # sub method replace the occurence of str1 with str2 in the given message
                message = re.sub('me','you',message)
                return message
  if 'my' in message:
                message = re.sub('my','your',message)
                return message
  if 'your' in message:
                message = re.sub('your','my',message)
                return message
  if 'you' in message:
                message = re.sub('you','me',message)
                return message
  return message
```

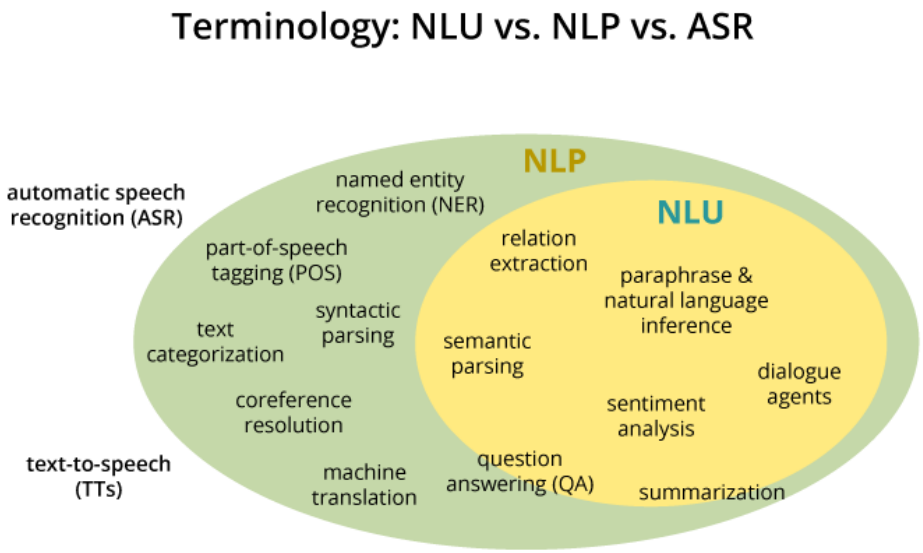
COMPLETE MODEL



INTENT RECOGNIZERS

NATURAL LANGUAGE UNDERSTANDING

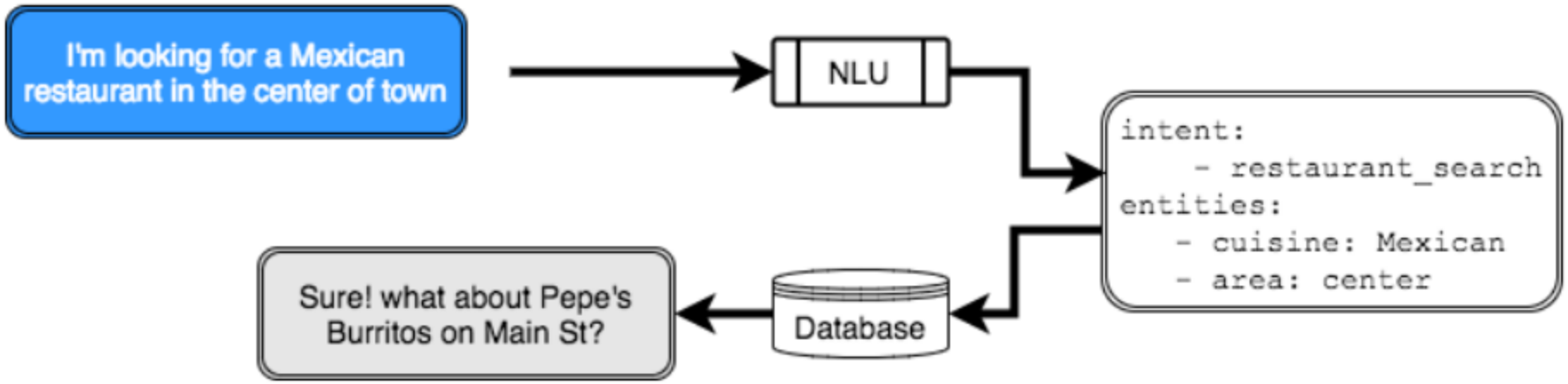




Intent: An intent represents the purpose of a user's input.

Entity: An entity represents a term or object that is relevant to your intents and that provides a specific context for an intent.

Dialog: A dialog is a branching conversation flow that defines responses to the defined intents and entities.

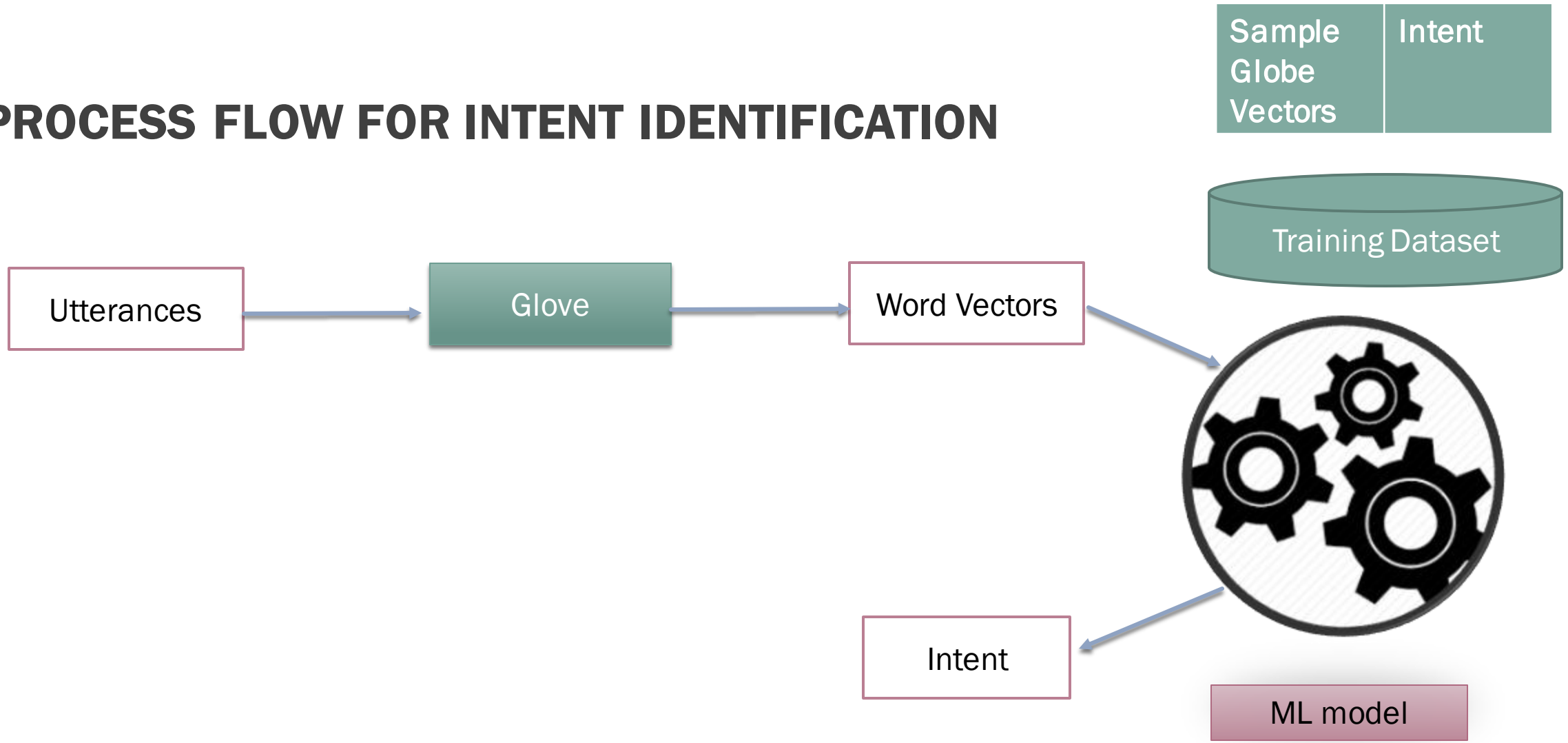


ATIS DATASET

This data set contains 4978 train and 893 test spoken utterances (text) classified into one of 26 intents.

Sample Utterances	Intent
BOS what is the arrival time in san francisco for the 755 am flight leaving washington EOS	flight_time
BOS find me the earliest flight from boston to atlanta EOS	flight
BOS what kinds of planes are used by american airlines EOS	aircraft
BOS please show me round trip tickets from denver to oakland EOS	airfare

PROCESS FLOW FOR INTENT IDENTIFICATION



CONVERSION OF TEXT INTO VECTORS (WORD EMBEDDINGS)

Theory of Distributional Hypothesis states that

Linguistic items with similar distributions (similar context) have similar meanings.

It was really cold yesterday.
It will be really warm today, though.
It'll be really hot tomorrow!
Will it be really cool Tuesday?

According to the Distributional Hypothesis, **the words cold, warm, hot and cool must be related in some way** (i.e., be close in meaning) because they occur in a **similar context, i.e., between the word "really" and a word indicating a particular day**. (Likewise, the words yesterday, today, tomorrow and Tuesday must be related, since they occur in the context of a word indicating a temperature.)

WORD VECTORS FORMED BY COUNTING CONTEXTS

It was the best of times, it was the worst of times.

Column – Context
"context" is just the word that precedes and the word that follows.

Row - Word

	START ___ was	it ___ the	was ___ best	the ___ of	best ___ times	of ___ it	times ___ was	was ___ worst	worst ___ times	of ___ END
it	1	0	0	0	0	0	1	0	0	0
was	0	2	0	0	0	0	0	0	0	0
the	0	0	1	0	0	0	0	1	0	0
best	0	0	0	1	0	0	0	0	0	0
of	0	0	0	0	1	0	0	0	1	0
times	0	0	0	1	0	0	0	0	0	1
worst	0	0	0	1	0	0	0	0	0	0

Number of times word has appeared in the context

GLOVE

GloVe is an unsupervised learning algorithm for obtaining vector representations for words.

Training is performed on aggregated **global word-word co-occurrence statistics** from a **corpus**, and the resulting representations showcase interesting linear substructures of the word vector space.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

The training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence. Owing to the fact that the logarithm of a ratio equals the difference of logarithms, this objective associates (the logarithm of) ratios of co-occurrence probabilities with vector differences in the word vector space.

ANALYZING WORD VECTORS OF GLOVE

1. To understand the contextual similarity between Glove vectors , **cosine similarity** measure is used.
2. Cosine SImilarity measures the difference In the angle between the vectors
   - 1 indicates that the vectors point in the same direction
   - 0 indicates they are perpendicular
   - -1 indicates they point in opposite directions

Difference between "cricket" and "football"

0.525491617392392 9

Difference between "foot" and "football"

0.2559363414015776

BUILDING SUPERVISED MODEL FOR INTENT RECOGNITION

Inference Using nearest neighbor classifier

1. Compute Glove vector for the test sentence – average of glove vectors for every word in the utterance

2. Calculate cosine similarity of test sentence with respect to all tuples in the training dataset

3. Intent for the test sentence will be same as that of intent of training sample with which it had the maximum similarity value

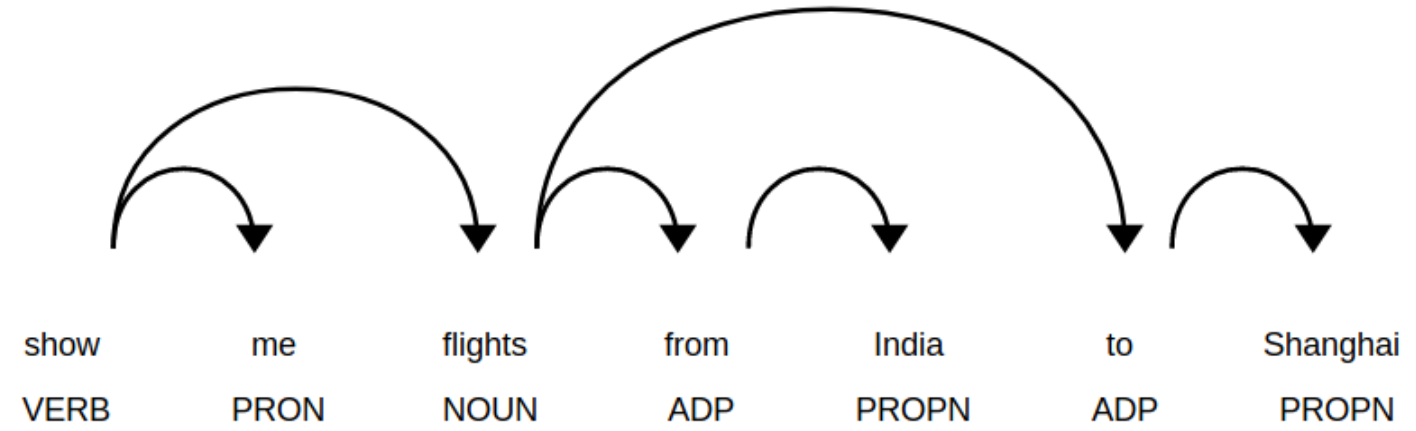
```
i would like to find a flight from charlotteto las vegas that makes
a stop in st. louis
```

Flight Information

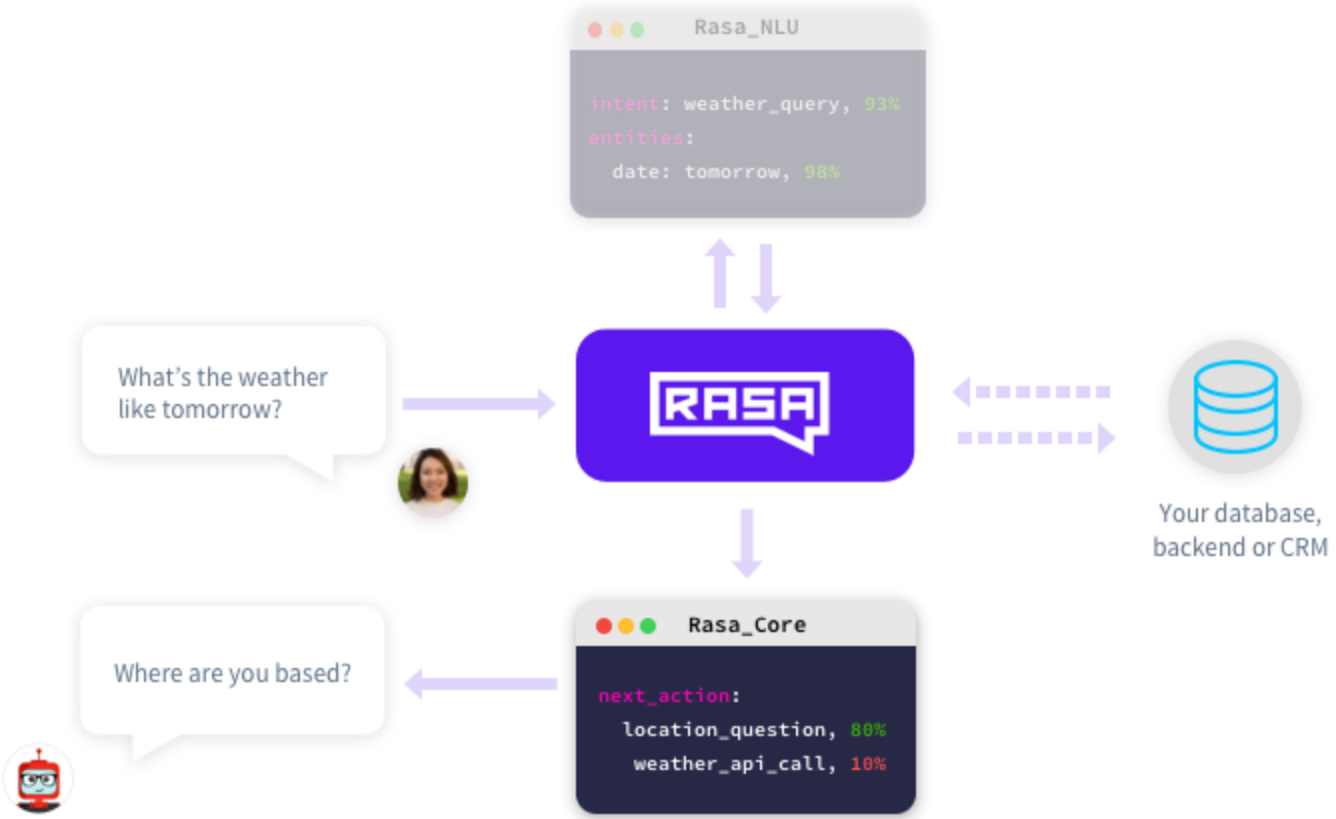
ENTITY RECOGNITION

- Keywords may not help in the identification of entities
- Contextual clues
  - Captilaization
  - Words occuring before & after
  - Numbers

Dependency parsing to identify the roles



VIRTUAL AGENTS (CREATING DIALOG FLOW (OR) LEVERAGING ON EXISTING API)


Rasa_NLU
intent: weather_query, 93%
entities:
date: tomorrow, 98%
What's the weather like tomorrow?
RASA
Your database, backend or CRM
Rasa_Core
next_action:
location_question, 80%
weather_api_call, 10%
Where are you based?

REFERENCES

1. https://onlim.com/en/the-history-of-chatbots/
2. https://rangle.io/blog/chatbots-an-introduction-to-conversational-ui/
3. https://medium.com/analytics-vidhya/building-a-simple-chatbot-in-python-using-nltk-7c8c8215ac6e
4. https://apps.worldwritable.com/tutorials/chatbot/
5. https://s3.amazonaws.com/assets.datacamp.com/production/course_3631/slides/chapter2.pdf
6. https://nlp.stanford.edu/projects/glove/
7. https://gist.github.com/aparrish/2f562e3737544cf29aaf1af30362f469#file-understanding-word-vectors-ipynb
8. https://github.com/santanu13/Weabot/blob/master/weabot.ipynb

Thanks for listening

Drop in a mail at nrr.it@psgtech.ac.in for queries

Check **ravitha.github.io** for my works and projects