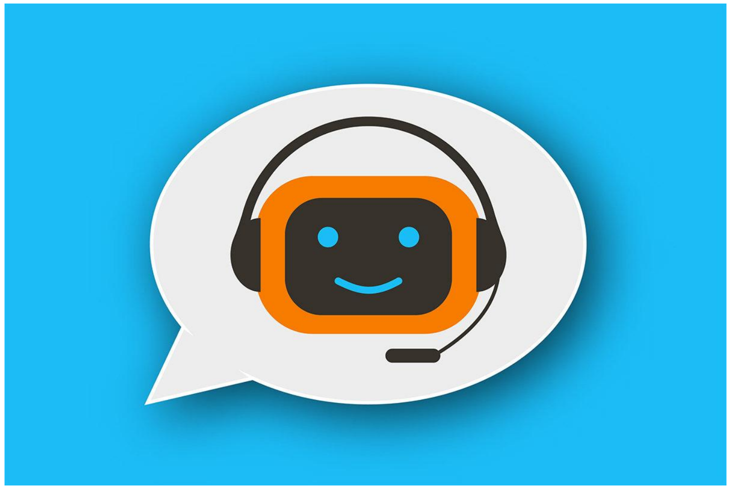
Attendance

Chatbots Introduction

What even is a chatbot?

What is a Chatbot?

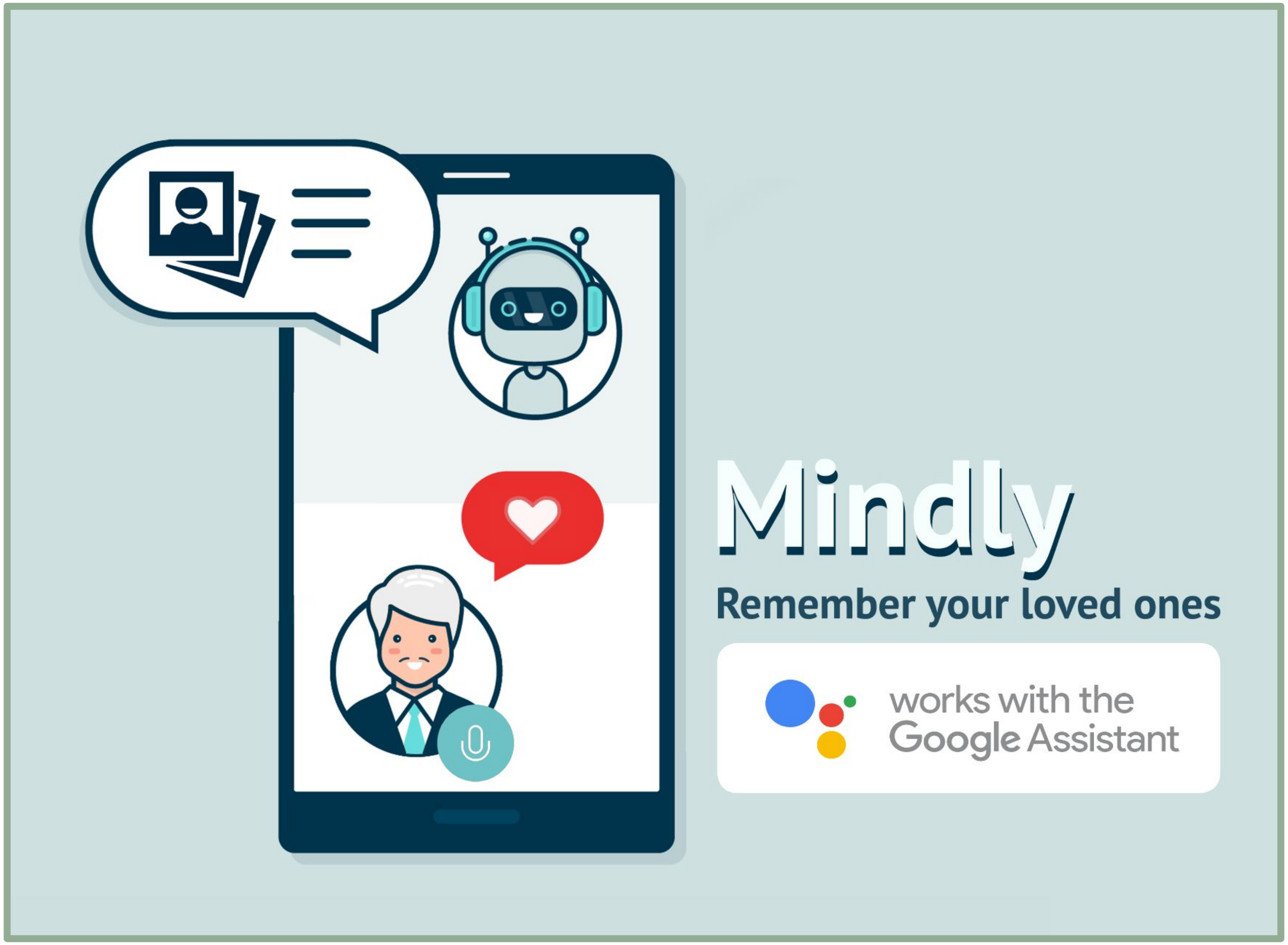
" A computer program designed to simulate conversation with human users, especially over the internet"



Common Types of Chatbots

- Question-Answer
  - Ask a question, it will answer it (factually) by extracting data from a corpus, such as a page of Wikipedia
- **Customer Support (what we're making!)**
  - Helps users with a set of predefined tasks the chatbot knows about
- Dialogue
  - Converses casually with the user
- Multilingual
  - Facilitates communication between users that speak different languages

Amazing Chatbot Application



“Customer Support” Chatbot Data

How does our chatbot know what to say?

What can Chatbots do?

The first thing we need to do is plan what our chatbot can do

For example, if we’re making a chatbot relating to **weather,** what should we be able to ask it to do?

Weather Bot Tasks

Some examples could be:

- ▷ Weather conditions (sunny, rainy, etc.)
- ▷ Daily temperature lows/highs
- ▷ Forecast for future days
- ▷ What to wear (shorts, jacket, etc.)

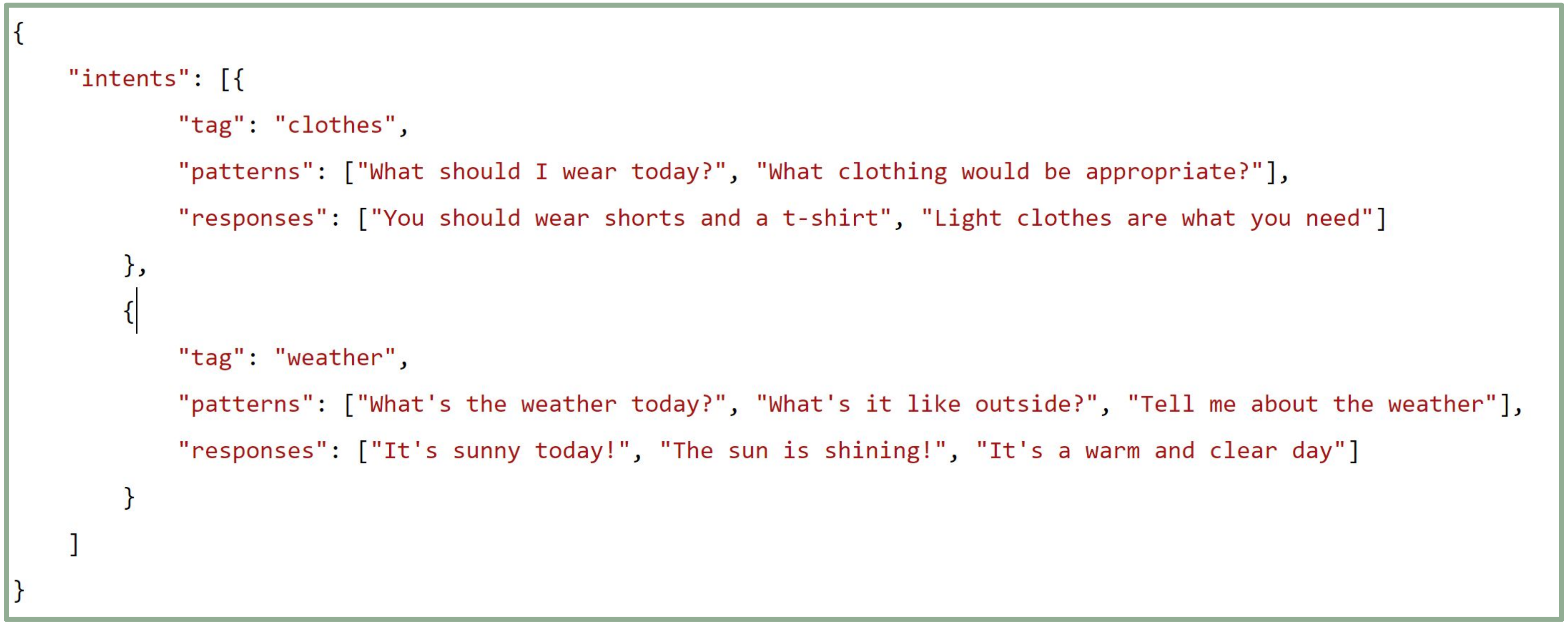
Categorizing Tasks

We’ll categorize these tasks into three categories: tag, patterns, and responses

- Tag: General summary for the task
- Patterns: User sentence inputs that relate to these tasks
- Responses: How the bot should respond to this input

Example File (Intents)

Below is some example data for our weather bot tasks:



```
{
    "intents": [{
            "tag": "clothes",
            "patterns": ["What should I wear today?", "What clothing would be appropriate?"],
            "responses": ["You should wear shorts and a t-shirt", "Light clothes are what you need"]
        },
        {
            "tag": "weather",
            "patterns": ["What's the weather today?", "What's it like outside?", "Tell me about the weather"],
            "responses": ["It's sunny today!", "The sun is shining!", "It's a warm and clear day"]
        }
    ]
}
```

The response values don’t have to be hard-coded. In this example, it would make more sense to get weather data from an external source, like an API

Tips for Customizing Intents

Some common tags I recommend in every simple chat bot:

- greeting
- goodbye
- thanks
- noanswer: for invalid user inputs
- options: tell the users what the bot can do

“Customer Support” Chatbot Model

How do we train our chatbot?

The Main Problem

We’ve learned about a couple of models that we can use to train our model:

- Linear Regression
- Logistic Regression
- Neural Networks

The problem with all of these models right now is that they all require numeric inputs, and we have words :(

Bag of Words Approach

A simple but effective approach we can use is the bag of words model

The main idea is to train a model with the **patterns** as the inputs and the **tag** as the output

Bag of Words Algorithm

In general, to create a bag of words we need to:

1. Tokenize and clean every word in the patterns
2. Add these words to a list that we'll call our bag
3. For the user input, create a list of 0s and 1s, where 1 means the word is in our bag, and 0 otherwise
4. Create an output vector corresponding to our tags

1) Tokenizing and Cleaning (review)

In this step, we make our words more consistent by:

- Removing punctuation
- Grouping similar words together
- Removing unnecessary words

For example:

```
"How was your day today"
```

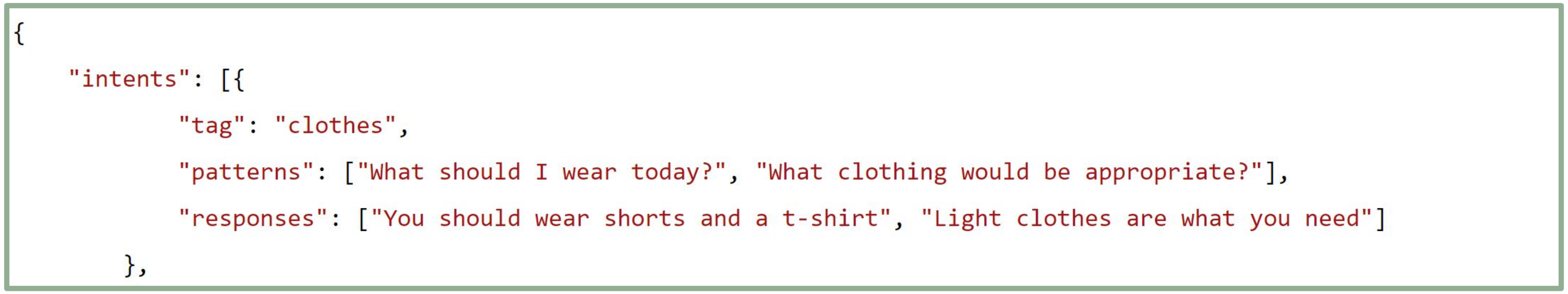
becomes

```
["how", "wa", "your", "day", "today"]
```

2) Creating our Bag

Our bag contains all unique words found in our patterns

If our weather bot data is:



```
{
    "intents": [{
            "tag": "clothes",
            "patterns": ["What should I wear today?", "What clothing would be appropriate?"],
            "responses": ["You should wear shorts and a t-shirt", "Light clothes are what you need"]
        },
```

Our bag would become:

```
[“what”, “should”, “wear”, “today”,
“clothing”, “be”, “appropriate”]
```

3) Numeric Data from User Inputs

User Input: `“What clothing for today?”`

Recall our bag from earlier:

```
[“what”, “should”, “wear”, “today”,
“clothing”, “be”, “appropriate”]
```

Numeric Data: `[1, 0, 0, 1, 1, 0, 0]`

For every word in the bag, if it exists in our user input, it becomes a 1, and a 0 otherwise

4) Output Vector

Our output vector for the tag must also be a numeric value!

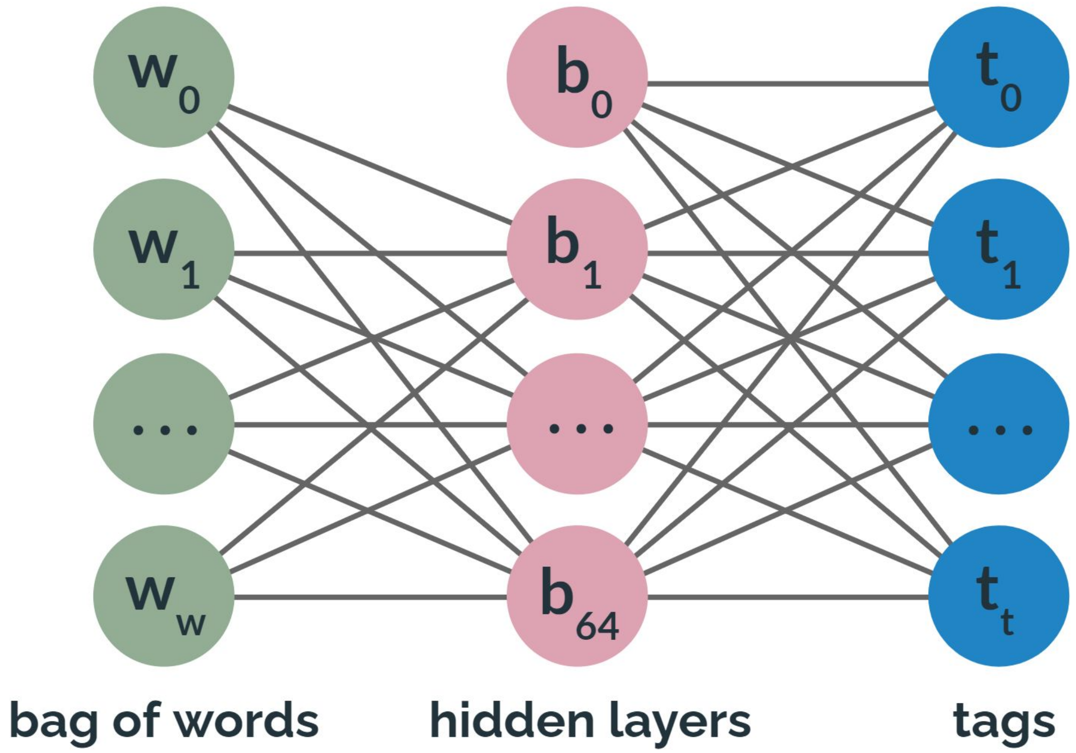
To accomplish this, we'll take a vector of zeros to be the same size as the number of tags. The element will be 1 when its index matches that of the tag

Tags: `["weather", "temperature", "clothes"]`

```
weather = [1, 0, 0]
temperature = [0, 1, 0]
clothes = [0, 0, 1]
```

Neural Network Training

Now that we have numeric inputs and outputs, we can train our model using a neural network!



(Don't worry I've coded all of this for you in the notebook)

Tasks to Complete

1) Work on the notebook (**chatbot_pt_1.zip)** in the google drive folder

   https://drive.google.com/file/d/1o72iwwzfvKVhSB9tI7f1nd85DO4YR6Nf/view?usp=sharing

Note: There's two notebook implementations, one that uses Keras/Tensorflow and one that doesn't. If you want to experiment with machine learning libraries, use the former. If you don't want to install any additional packages, use the latter

Try to work on it collaboratively! You might meet some people you could do a project with in the future

As always, let us know if you need any help!

SIG
NLL