# chatbot

# What is chat bot???

A chatbot is artificial intelligence (AI) software that can imitate a natural language discussion (or chat) with a user via messaging apps, websites or mobile apps.

# What is the significance of chat bot???

- **A chatbot is frequently described as one of the most advanced and promising forms of human-machine interaction.**
- **Chatbots can automatically simulate interactions with customers based on a set of predefined conditions or event**

**From a technology standpoint, however, a chatbot is simply the next step in the evolution of a Question Answering system that uses Natural Language Processing (NLP).**

- **One of the most common examples of Natural Language Processing is usage in many organizations' end-use applications in formulating responses to inquiries in natural language.**

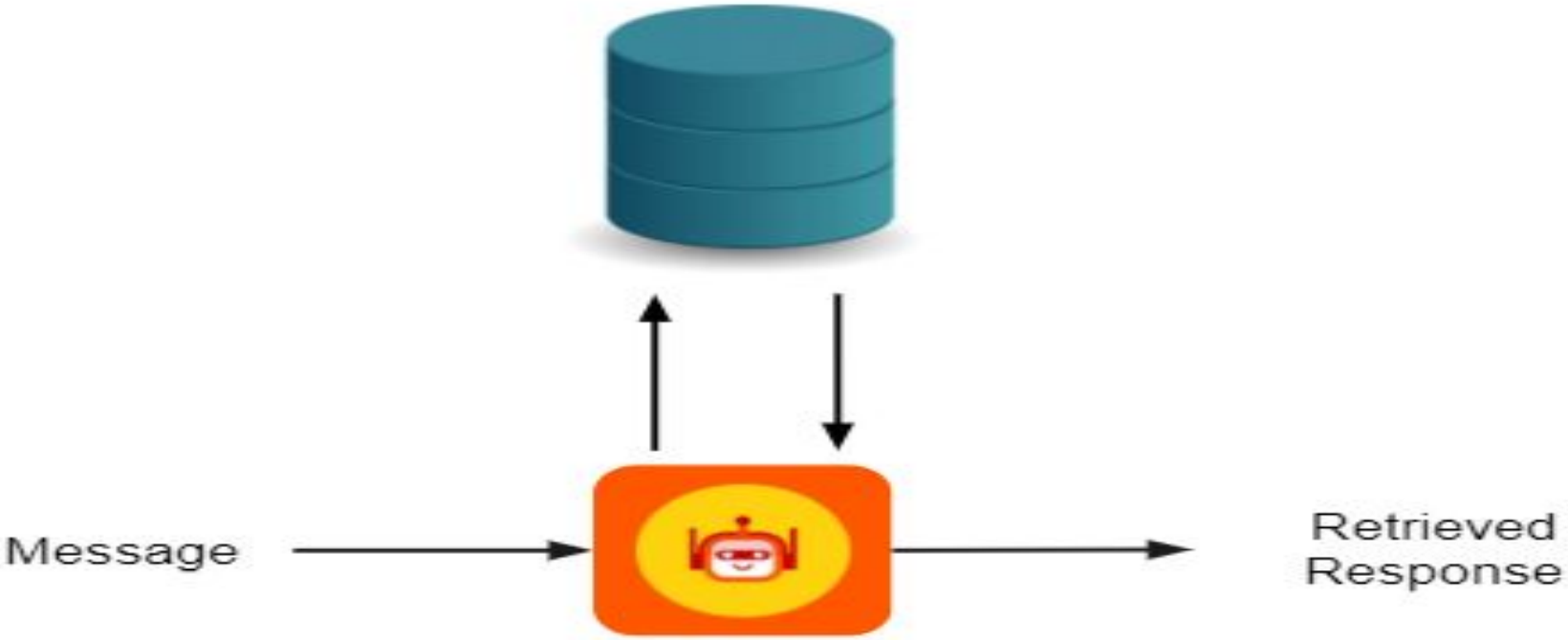
# Types of chat bot

- **Generative Based chatbot**

- **Retrieval based chatbot**

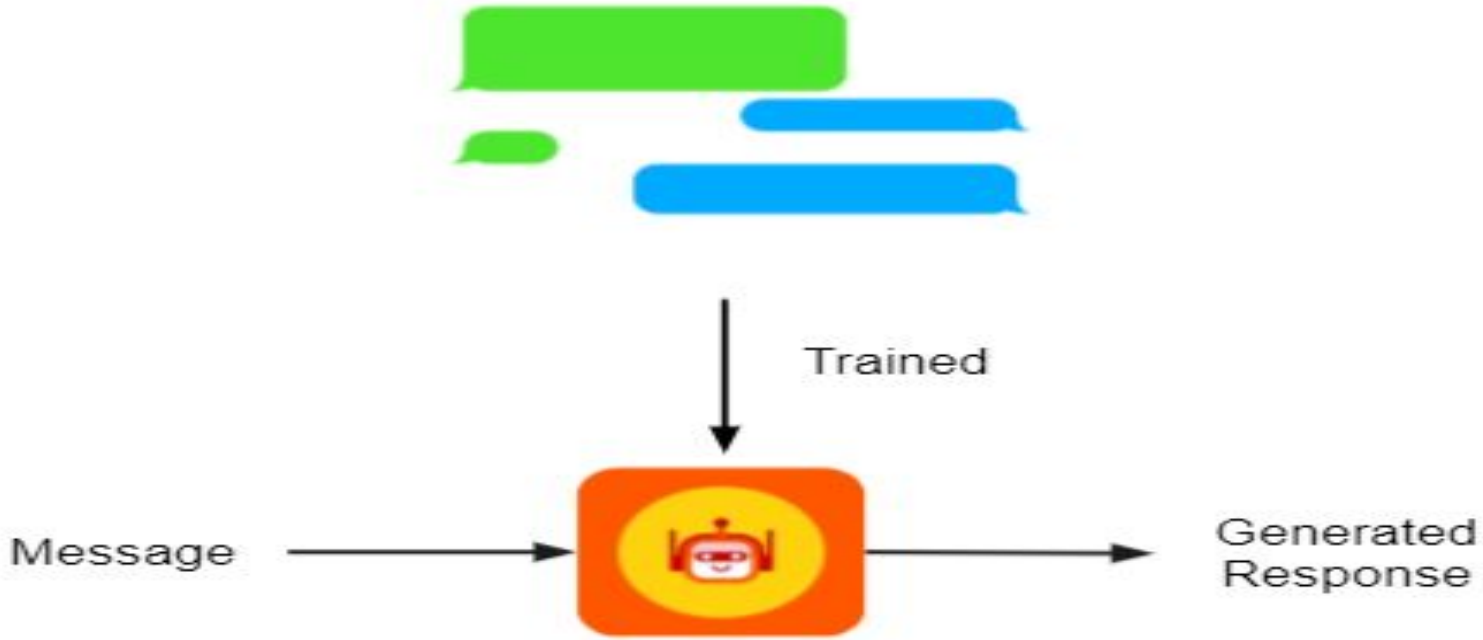
# Retrieval based chatbot

- **Retrieval based chatbots, employ techniques such as keyword matching, machine learning, and deep learning to find the most appropriate response.**
- **These chatbots, regardless of technology, solely deliver predefined responses and do not generate fresh output.**
- **From a database of predefined responses, the chatbot is trained to offer the best possible response.**
- **The responses are based on previously collected data.**


Message
Retrieved
Response

# Generative based chatbot

- **Generative chatbots use a combination of supervised learning, unsupervised learning & reinforcement learning.**
- **A generative chatbot is an open-domain chatbot that creates unique language combinations rather than selecting from a list of predefined responses.**
- **Retrieval-based systems are limited to predefined responses.**
- **Chatbots that use generative methods can generate new dialogue based on large amounts of conversational training data.**


Trained
Message
Generated
Response

# Natural language process

Natural language processing is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data.

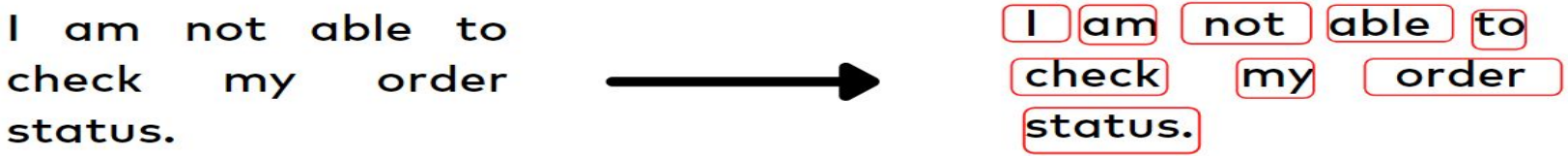
## importing libraries and modules

Import nltk and numpy libraries for the text preprocessing.

```
import nltk

import numpy as np
```

# Tokenization

Tokenization is the process of breaking down sentence or paragraphs into smaller chunks of words called tokens.



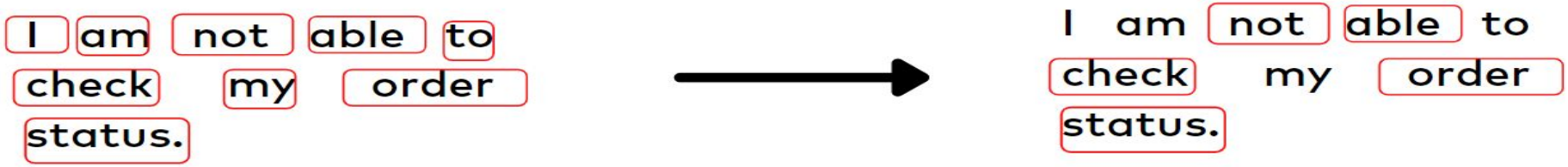

Tokenizing a sample query from a user

```
nltk.download('punkt') # downloading model to tokenize message
from nltk.tokenize import word_tokenize
```

Downloading 'punkt' model to tokenize message and importing 'word_tokenize' for word tokenization.

# Stopword removal

On removal of some words, the meaning of the sentence doesn't change, like and, am. Those words are called stop-words and should be removed before feeding to any algorithm. In datasets, some non-stop words repeat very frequently. Those words too should be removed to get an unbiased result from the algorithm.




```
nltk.download('stopwords') # downloading stopwords
```

```
from nltk.corpus import stopwords
```

# Lemmatization

**Lemmatization is the process of converting a word to its base form. It considers the context and converts the word to its meaningful base form, whereas stemming just removes the last few characters, often leading to incorrect meanings and spelling errors.**

**For example, lemmatization would correctly identify the base form of ‘moving’ to ‘move’.**
**‘Moving’ -> Lemmatization -> ‘Move’**
Downloading all the lemmas present in English Language & WordNetLemmatizer for Lemmatization.

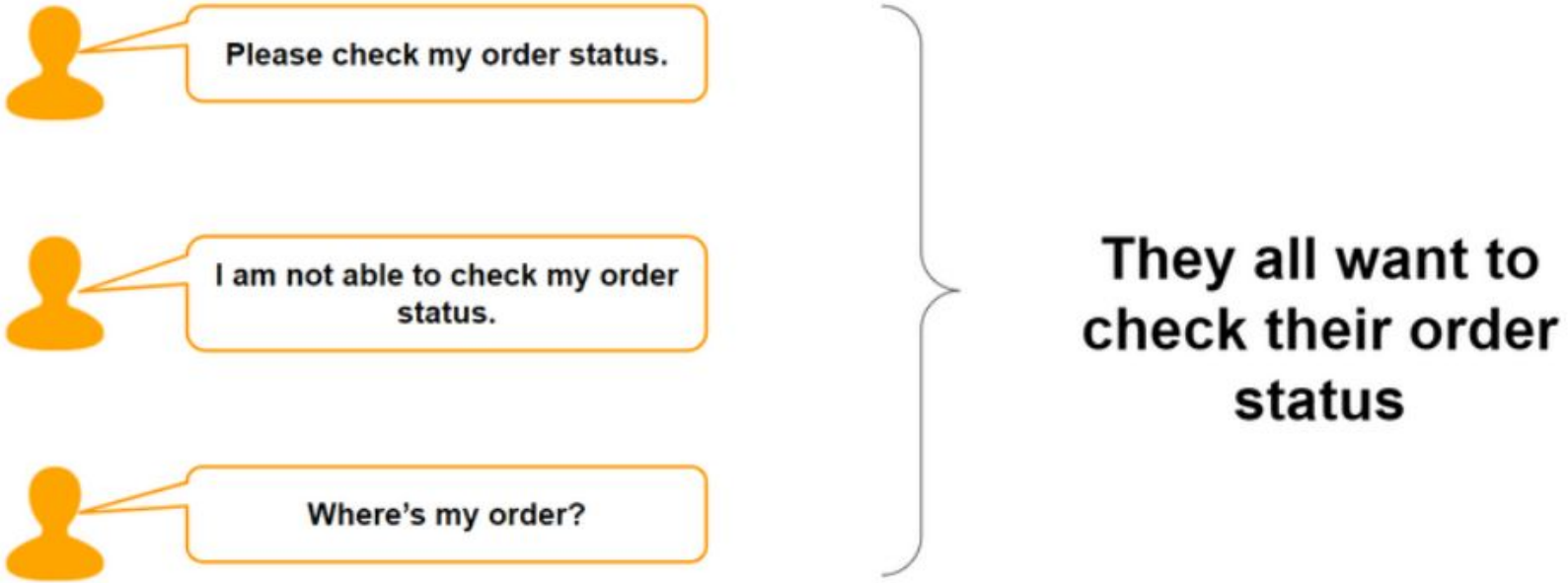
```
nltk.download('wordnet') # downloading all lemmas of english language

from nltk.stem import WordNetLemmatizer
```

# VECTORIZATION

- After tokenization, and stop words removal, our "content" are still in string format.
- We need to convert those strings to numbers based on their importance (features).
- We use TF-IDF vectorization to convert those text to vector of importance.
- With TF-IDF we can extract important words in our data.
- It assign rarely occurring words a high number, and frequently occurring words a very low number.
- We can define our own function for stop-words removal, tokenization & lemmatization.
- It takes the corpus of words as an argument.

# INTENT

- Intent recognition is a form of natural language processing (NLP), a subfield of artificial intelligence.
- Natural language processing (NLP) and analysis is concerned with computers processing and analyzing natural language, that is, any language that has evolved naturally rather than artificially, such as computer coding languages.
- Intent classification or intent recognition is the task of taking a written or spoken input, and classifying it based on what the user wants to achieve.
- Intent recognition forms an essential component of chatbots and finds use in sales conversions, customer support, and many other areas.




In the above example, three different users are inquiring about their order status using different sentences but the underlying intent is to check what's the current status of their order.

# LOADING THE INTENT FROM THE “INTENT.JSON”FILE

```
import json

with open('/content/intents.json') as file:

    intents = json.load(file)
```

This is how your json file looks. It needs to be cleaned and stored in the form of the vectors.


```
{
        "tag": "greeting",
        "patterns": ["Hi there", "Is anyone there?","Hey","Hola", "Hello", "Good day"],
        "responses": ["Hello, how can I help?", "Good to see you again", "Hi there, how can I help?"]
},
{
        "tag": "goodbye",
        "patterns": ["Bye", "See you later", "Goodbye", "Nice chatting to you, bye", "Till next time", "No, that's it"],
        "responses": ["See you!", "Have a nice day", "Bye! Come back again soon."]
},
{
        "tag": "thanks",
        "patterns": ["Thanks", "Thank you", "That's helpful", "Awesome, thanks", "Thanks for helping me"],
        "responses": ["Happy to help!", "Any time!", "My pleasure"]
},
```

# END