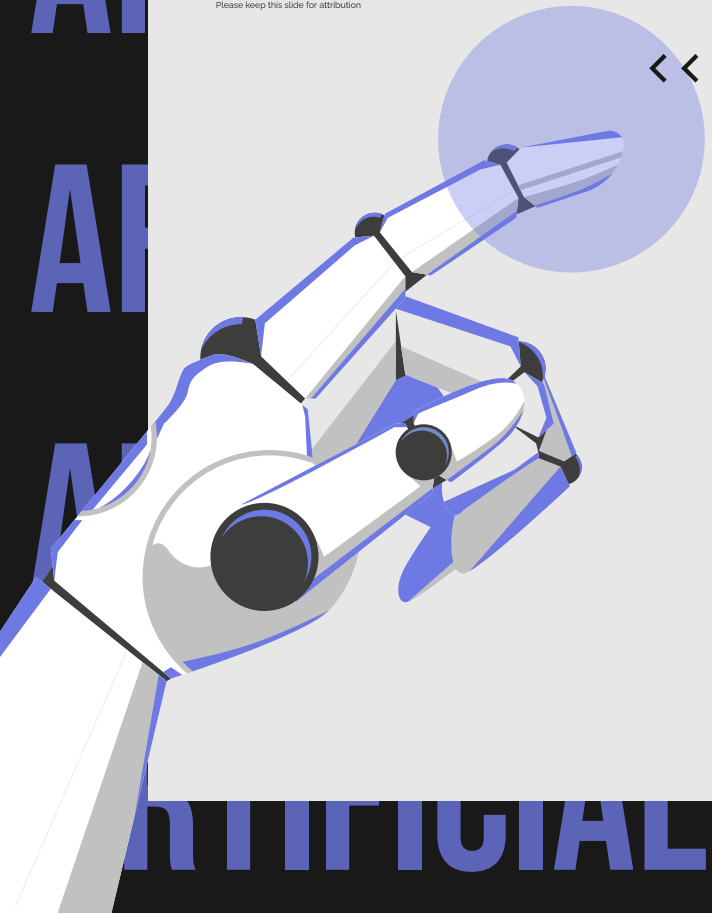


CHATBOT - CHATTY

Intelligent Systems II

2023

Daniela Dias, 98039
Filipe Gonçalves, 98083
Gonçalo Machado, 98359
João Borges, 98155
Miguel Beirão, 98157

TABLE OF CONTENTS

ARTIFICIAL INTELLIGENCE (AI)

01.

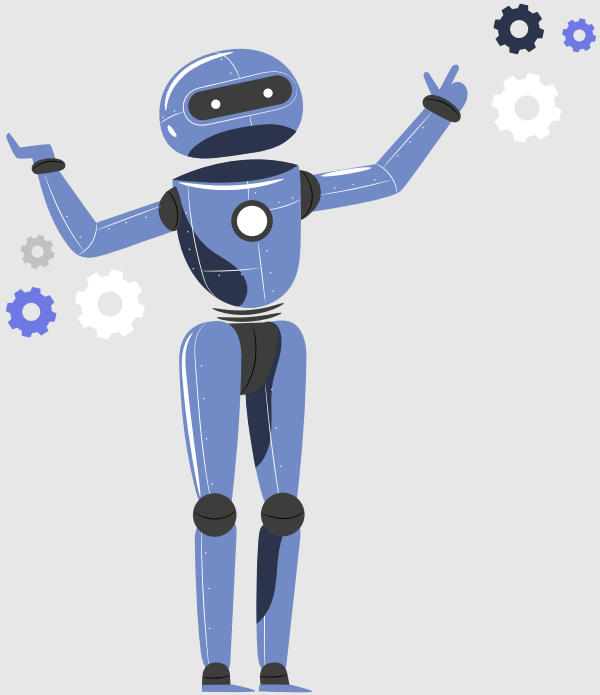
INTRODUCTION

Brief contextualization and introduction

INTRODUCTION

Conversational agents, sometimes known as chat bots, have several applications, namely for remote automatic assistance.

On the other hand, if a conversational agent has a conversational behavior that makes it indistinguishable from the behavior expected in a human being, it may be considered that this agent has human-level intelligence, as Alan Turing suggested.



INTRODUCTION

The proposed work consists in the development of a conversational agent with:

- Natural language processing for some common sentences types
- Ability to accumulate knowledge provided by interlocutors and produce answers to questions
- React in an intelligent way for any malformed or grammatically incorrect sentences

/[AI]

02.

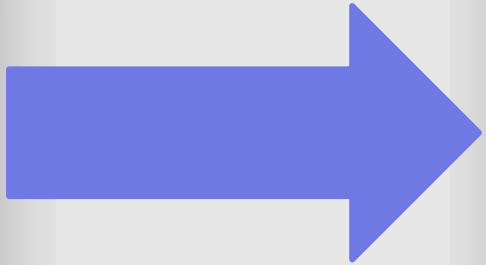
DATASET

Explanation of used dataset and its benefits

DATASET

```
{
 "intent": "Greeting",
 "text": [...],
 "responses": [...],
 "extension": {
   "function": "",
   "entities": false,
   "responses": []
 },
 "context": {...},
 "entityType": "NA",
 "entities": []
}
```

CHANGES



```
{
 "intent": "Greeting",
 "text": [
    {
     "english": [...],
     "portuguese": [...]
    }
 ],
 "responses": [
    {
   "english": [...],
   "portuguese": [...]
    }
 ],
 "entities": []
 ...
}
```

DATASET FINAL

```
{
    "intent": "Greeting",
    "text": [
        {
            "english": [
                "Hi",
                "Hi there",
                "Hola",
                "Hello",
                "Hello there",
                "Hya",
                "Hya there",
                "Hola human, please tell me your GeniSys user"
            ],
            "portuguese": [
                "Olá",
                "Ei",
                "Oi",
                "olá"
            ]
        }
    ],
    "responses": [
        {
            "english": [
                "Hi human, please tell me your GeniSys user",
                "Hello human, please tell me your GeniSys user",
                "Hola human, please tell me your GeniSys user"
            ],
            "portuguese": [
                "Olá humano, por favor indique o seu GeniSys user",
                "Ei humano, por favor indique o seu GeniSys user"
            ]
        }
    ],
    "entities": []
},
```

03.

TOKENIZER

How we pre-process text for further analysis

TOKENIZER

In Natural Language Processing (NLP), tokenization is often the first step in text pre-processing.

TOKENIZATION:

- The process of breaking down a sentence into individual tokens (either words, characters, or subwords).

MAIN GOAL:

- Provide a structured representation of text data that can be used for further analysis or processing.

TOKENIZER

Our implementation tokenizes a text string into a dictionary of tokens and their possible words.

OUR IMPLEMENTATION

- **Split a sentence into individual words.**
- **Remove punctuation and special characters.**
- **Remove accents.**
- **Ignore words that don't meet a minimum length.**
- Remove stop words.
- **Normalize words to lowercase.**
- **Stem words.**
- Lemmatize words.

Bold - Default Tokenizer

STEMMER VS LEMMATIZER

STEMMER

- Stems or removes last few characters from a word.
- E.g. "history" and "historie" (misspelling) would be under the key "histori".

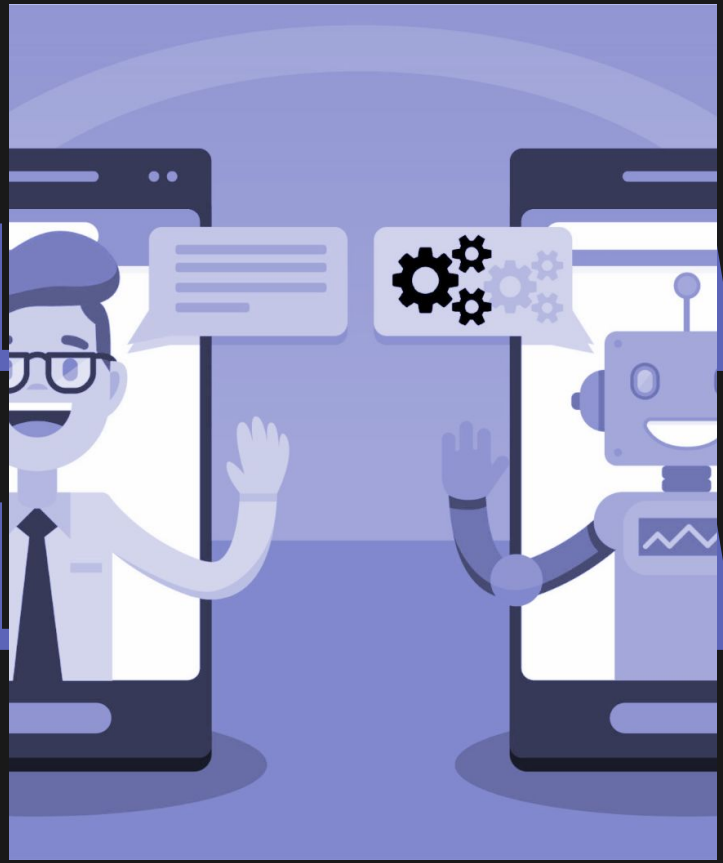
- Considers the context and converts the word to its meaningful base form (called Lemma).
- E.g. “caring” would be under the key “care”.

LEMMATIZER

STEMMER VS LEMMATIZER

From our experiments, we found that the best results are obtained by using **stemming** instead of lemmatization.

With stemming, we can handle more forms of user input, such as **misspellings** or **variations of words**.



DETECT LANGUAGE

Our ChatBot has natural language processing (NLP) for both Portuguese and English.

DETECTION ATTEMPT:

- Count how many tokens are recognized for each language.
- Select the language with more recognized tokens.

FAILED ATTEMPT:

- For all languages, check misspellings for unrecognized tokens with, at least, **3 letters**.
- Obtain tokens that are, at least, **60% similar** (suggested corrections).
- Select the language with more recognized tokens after spelling check.

04.

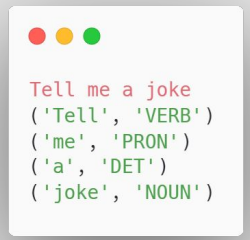
GRAMMAR DETECTION

How do we detect malformed sentences

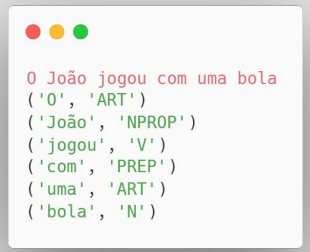
TAGGER

To reduce the size and the clutter of our grammar, we decided to not include what class each word belongs to (Eg. eat -> verb).

To do this, we used **taggers**, which tag words with their class. We used 2 pre-trained taggers, one for english that is provided by **nltk**, and another for portuguese that we found on Github.



English Tagger



Portuguese Tagger

GRAMMAR

```
S -> NOUN
S -> ADJT NOUN
S -> NP VP
S -> VP
S -> S CONJ S
PP -> P NP
NP -> DT NP | NOUN PP | NOUN | ADJT NOUN
| ADVB ADJT NOUN | ADVB ADJT | DT NOUN | P | PREPP NOUN | NOUN NP
VP -> VERB NP | VERB PP | VERB NP PP | VERB
ADJT -> 'ADJ'
ADVB -> 'ADV' | 'ADV-KS' | 'ADV-KS-REL'
DT -> 'ART'
NOUN -> 'N' | 'NPROP'
P -> 'PROADJ' | 'PRO-KS' | 'PROPESS' | 'PRO-KS-REL' | 'PROSUB'
PREPP -> 'PREP' | 'PREP|+'
VERB -> 'V' | 'VAUX'
CONJ -> 'KS' | 'KC'
EST -> 'N|EST'
```

Portuguese Grammar

```
S -> N
S -> ADJT N
S -> NP VP
S -> VP
PP -> P NP
NP -> DT NP | N PP | N | ADJT N
| ADVB ADJT N | ADVB ADJT | DT N | P | N NP
VP -> V NP | V PP | V NP PP | V | V PRTT NP | V VP
ADJT -> 'ADJ'
ADVB -> 'ADV'
DT -> 'DET'
N -> 'NOUN'
P -> 'PRON'
V -> 'VERB'
PRTT -> 'PRT'
X -> 'X'
```

English Grammar

ARTIFICIAL INTELLIGENCE (AI)

GRAMMAR CHECKER

```
Chatty: Hello, I am Chatty. How can I help you ?
        Olá, eu sou o Chatty. Como posso ajudar ?
> Give me joke a
Chatty: You should check your grammar!
         Devias verificar a tua gramática!
```

Grammatically incorrect sentence

```
Show me the semantic tree
Chatty: Here is the semantic tree from the last sentence:Tell me a joke
          S
          |
         VP
  ________|___
 |            PP
 |     _______|___
 |    |           NP
 |    |         ___|___
 V    P        DT      N
 |    |        |       |
Tell  me       a      joke
```

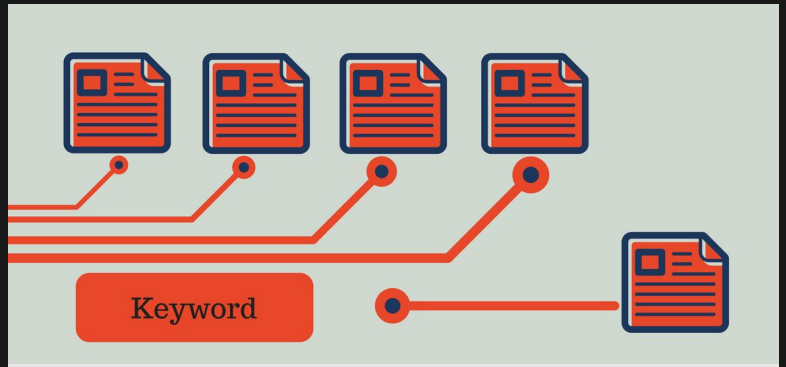
Sentence Sintatic Tree

ARTIFICIAL INTELLIGENCE (AI)

05.

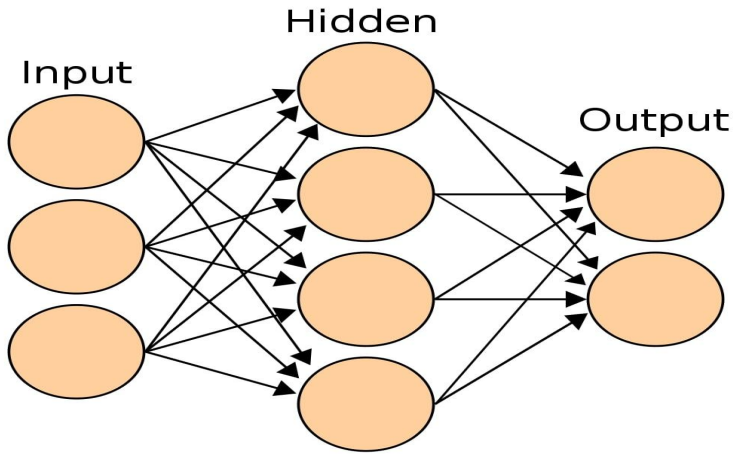
MACHINE LEARNING MODEL

What model was used and how it was trained



TF-IDF VECTORIZER

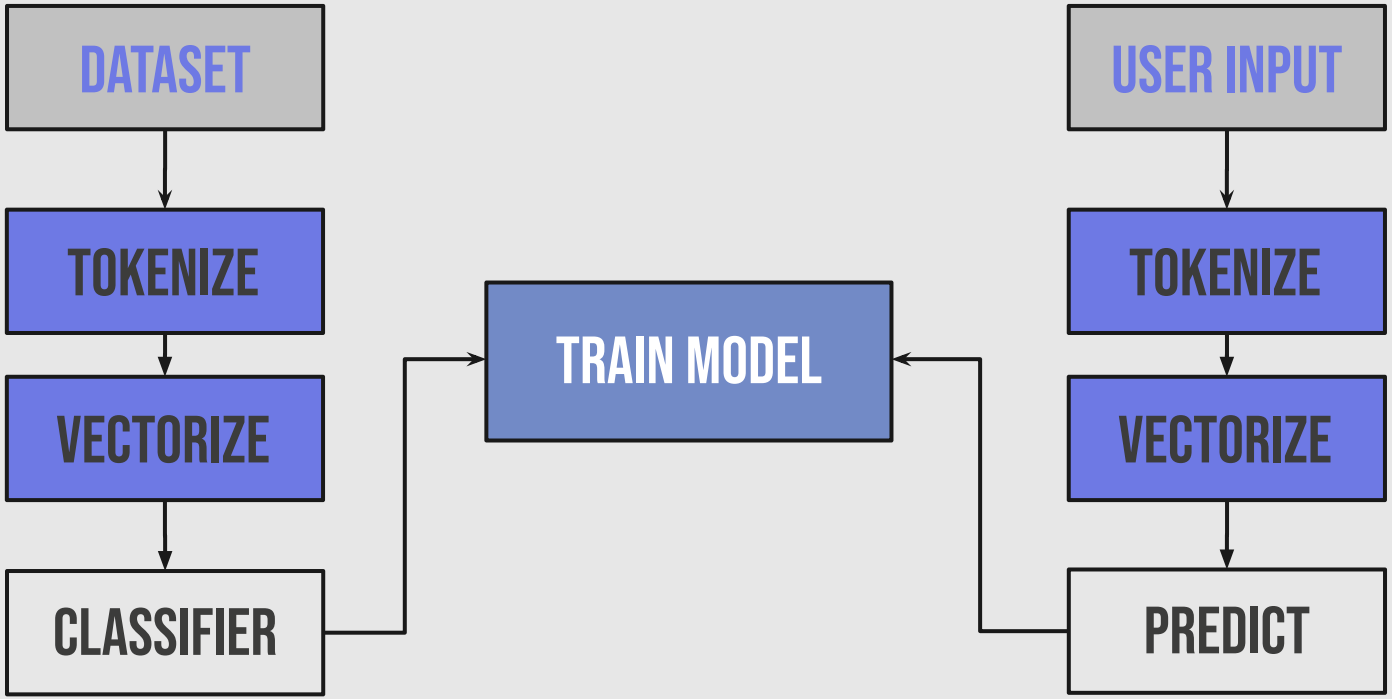
Vectorizer to transform the user input into a matrix based on word importance in the sentence



MULTI-LAYER PERCEPTRON CLASSIFIER

Artificial neural network model to map input data to a set of appropriate outputs

ORGANIZATIONAL CHART



[AI]

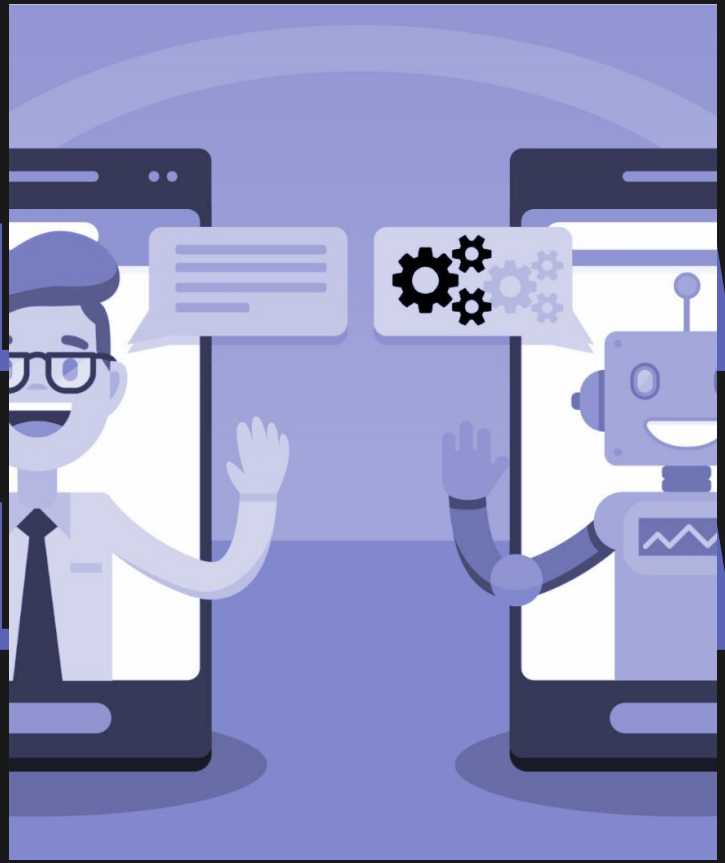
06.

CONTEXT LEARNING

How the bot learns from the conversations

CONTEXT LEARNING

Chatty is able to remember information that is given by the user and use it according to the context given, this is done by using **spaCy** and other methods.



USING SPACY

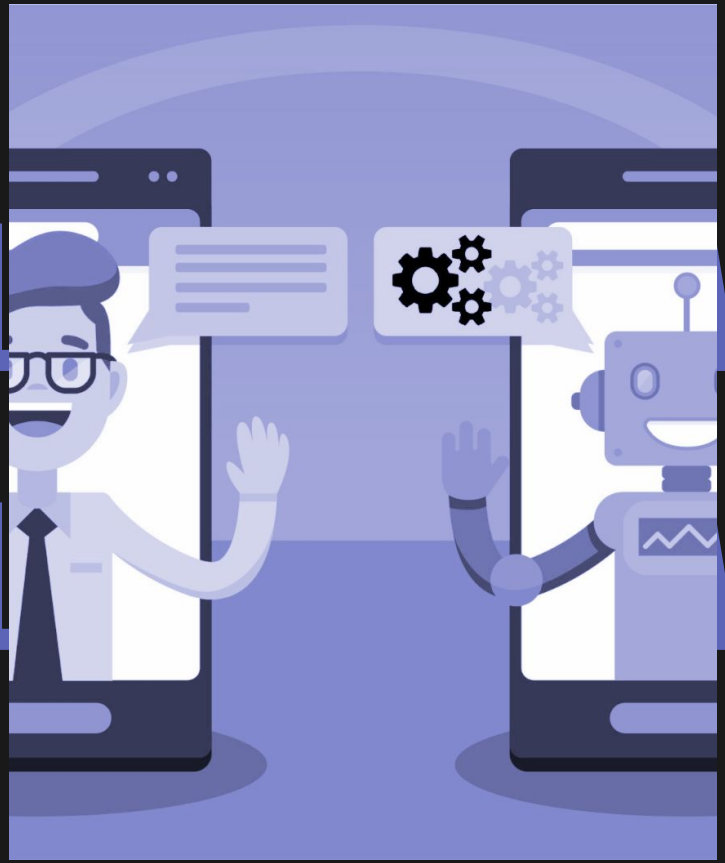
- To recognize entities, it loads a trained pipeline for the English language
- We then try to extract entities in the prompt given by the user
- We convert the prompt using the nlp() function, to extract meaning from the human language for the computer, which uses the trained pipeline referred previously
- This will return all the entities present in the prompt and a tag related to them, such as PERSON or COUNTRY

OTHER METHOD

- As the trained pipeline only works for english entities, and does not recognize objects like fruits, we developed another workaround to store the information
- When a word is not present in any database, we replace the word with the tag <NULL>, present in all the prompts in the dataset, to check if there is any matching intents present in the database
- If there is, we will save the word replaced, and then save it to the entities

LEARN FROM MISTAKES

We also implemented a “Learn from my mistakes” feature.
When the chatbot recognises that its response wasn't what the user wanted/expected, it will remember a new answer, provided by the user.



FUTURE WORK

FUTURE WORK

Although Chatty already talks fluently, it still needs some improvements, such as:

- The chatbot can only remember one thing from a prompt given by the user. If I say “my name is John and i'm 20 years old”, it will only remember that “i'm 20 years old”.
- Improve the machine learning and study the best activation function for the hidden layers.
- Improve both grammars to accept a more widen variety of sentences.
- Broaden the dataset so that includes more intents and possibilities of responses.
- Have a real dictionary as a base of tokens to include in the model, and not only have tokens from responses and questions.