

Conversational ChatBot - Chatty

Filipe Gonçalves
DETI
MRSI
Aveiro, Portugal
98083

Gonçalo Machado
DETI
MEI
Aveiro, Portugal
98359

João Borges
DETI
MEI
Aveiro, Portugal
98155

Daniela Dias
DETI
MEI
Aveiro, Portugal
98039

Miguel Beirão
DETI
MEI
Aveiro, Portugal
98157

Abstract—Conversational agents, sometimes known as chatbots, 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. The proposed work consists in the development of a conversational agent with the following characteristics: - Natural language processing (Portuguese and/or English) for some common sentences types; - Ability to accumulate information/knowledge provided by interlocutors (i.e. learn from interaction) and produce answers to questions; - For grammatically incorrect sentences, or sentences not supported by the system, react in a "seemingly intelligent" way.

Index Terms—Machine Learning, nltk, Stemmer, chatbot, NLP, TF-IDF

I. INTRODUCTION

Chatbots are currently trending, mainly because of the newest and most utilized chatbot, ChatGPT, and the various variations and counter measures already implemented.

We decided to develop a chatbot that thinks and articulates a response based on the intent of the message provided by the user.

Firstly, we decided to find a good dataset to use. With this dataset, we can transform the words into more usable identifiers (tokens), train the model and, finally, let the chatbot run wild.

As for what we implemented, we will now list all the features we currently have:

- Text comprehension for both English and Portuguese
- Tokenization of words
- Language detection
- Language correction
- Grammar checker
- Syntactic Tree Visualization
- Message prediction
- Vectorize message based on word importance
- Entity detection and substitution
- Learning with mistakes corrected by the user
- Learning with every user input
- Save the intents to use afterwards

II. DATASET

A proper dataset is required for any AI. With this in mind, we decided to use one already available, instead of making one ourselves. This gave us less work (and a head start), allowing us to create a more complex and complete dataset.

After some research, we agreed to use the "simple-chatbot dataset" [1] as our base dataset and we expanded it as we went along with the project. This dataset consists of 22 different intents, each with a different number of questions the user could ask the chatbot and answers the user could receive. It also contains extra tags that could prove useful for more complex questions and answers, as we can see in Figure 2.

```
{
  "intents": [
    {
      "intent": "Greeting",
      "text": [
        {
          "Hello",
        }
      ],
      "responses": [
        {
          "Hi human, please tell me your Chatty user",
        }
      ],
      "extension": {
        "function": "",
        "entities": false,
        "responses": []
      },
      "context": {
        "in": "",
        "out": "GreetingUserRequest",
        "clear": false
      },
      "entityType": "NA",
      "entities": []
    },
    {
      ...
    }
  ]
}
```

Fig. 1. Initial Dataset

With the base dataset decided, we started by translating the text of both questions and answers so that the chatbot could communicate in English and Portuguese. We also removed topics that did not seem to be of interest, whilst adding new ones.

After the initial tests were completed and the results proved fruitful, the next step was to streamline the dataset. This was done by removing tags, such as "entityType", or by changing them, just as "function" in "extensions".

```

{
  "intents": [
    {
      "intent": "Greeting",
      "text": [
        {
          "english": [
            "Hello",
          ],
          "portuguese": [
            "Olá",
          ]
        }
      ],
      "responses": [
        {
          "english": [
            "Hi human, please tell me your Chatty user",
          ],
          "portuguese": [
            "Olá humano, por favor indique o seu Chatty user",
          ]
        }
      ],
      "extension": {
        "function": "",
        "entities": false,
        "english": [],
        "portuguese": []
      },
      "context": {
        "in": "",
        "out": "GreetingUserRequest",
        "clear": false
      },
      "entities": []
    },
    {
      ...
    }
  ]
}

```

Fig. 2. Dataset with some changes

Finally, with the bot being able to learn from its interactions, we decided to create a simple dataset that would update with every conversation, resulting in the final dataset. It would only possess the minimal amount of information that the bot needed and have both the intents from the previous datasets as well as the new interactions. We can see this in the Figure 3.

```

{
  "intents": [
    {
      "intent": "Greeting",
      "text": [
        {
          "english": [
            "Hello",
          ],
          "portuguese": [
            "Olá",
          ]
        }
      ],
      "responses": [
        {
          "english": [
            "Hi human, please tell me your Chatty username",
          ],
          "portuguese": [
            "Olá humano, por favor indique o seu Chatty username",
          ]
        }
      ],
      "entities": [],
      "extension": {
        "function": ""
      }
    },
    {
      ...
    }
  ]
}

```

Fig. 3. Final Dataset

III. TOKENIZER

Tokenization is the process of breaking down a sentence into individual tokens (either words, characters, or subwords). The main goal of tokenization is to provide a structured representation of text data that can be used for further analysis or processing.

In Natural Language Processing (NLP), tokenization is often the first step in text pre-processing. The text pre-processing tasks may include tasks such as stop word removal, stemming, lemmatization, sentiment analysis, named entity recognition, and many others. These tasks all require tokenization as a first step, as they operate on individual tokens rather than on the entire text string. It is important to note that the way in which the text is tokenized can have a significant impact on the performance of the natural language processing algorithm.

There are various approaches to tokenization, such as simple tokenization that separates text by whitespace or punctuation, or more complex approaches that take into account syntactic and semantic information. The choice of tokenization approach depends on the specific application and the language being used.

In the context of building a ChatBot, tokenization is a critical step in understanding user input and generating an appropriate response. The ChatBot needs to identify the individual words (or tokens) in the user's input in order to determine what the user is asking or saying. Once the input has been tokenized, the ChatBot can apply various natural language processing techniques to analyze the input.

For example, if a user inputs the sentence "I want to book a flight to New York", the ChatBot needs to understand that the user wants to book a flight to New York. Tokenization can help identify the individual tokens in the sentence, such as "I", "want", "to", "book", "a", "flight", "to", "New", and "York", which can then be used to determine the user's intent and generate an appropriate response.

Tokenization also helps the ChatBot handle different forms of user input, such as misspellings or variations of words. For example, if a user types "runing" instead of "running", tokenization can still identify the word "running" as a token and the ChatBot can understand what the user is trying to say.

Our implementation tokenizes a text string into a dictionary of tokens and their possible words (e.g., "jumping" and "jumped" would both be under the key "jump" if stemming is used). For this, our Tokenizer first splits the input text into individual words, removes punctuation and special characters, removes accents, and applies the specified token size and stop-word filters. It then normalizes each token to lowercase and applies stemming and/or lemmatization if specified [9].

We provide a flexible and customizable way to tokenize text, which allows us to specify the minimum token size, the language for stop-words, and the type of token normalization. In summary, our Tokenizer takes four optional arguments:

- `min_token_size`: An integer representing the minimum size of a token to be considered. If a token has fewer characters than this, it will be ignored. If not specified, all tokens will be considered.
- `language`: A string representing the language to use for stop-words. If not specified, no stop-words will be removed.
- `stemmer`[13]: A stemmer object from the `nlTK.stem` module to use for stemming tokens, in other words, it removes

the last few characters from a word. If not specified, no stemming will be performed.

- **lemmatizer**[14]: A lemmatizer object from the `nlk.stem` module to use for lemmatizing tokens, in other words, it considers the context and converts the word to its meaningful base form (called Lemma). If not specified, no lemmatization will be performed.

From our experiments, we found that the best results are obtained by using stemming instead of lemmatization, by using a minimum token size of 2 and by keeping all stopwords.

IV. LANGUAGE DETECTION

To determine whether a user is speaking in Portuguese or English, our chatbot uses natural language processing (NLP) for both languages. This involves counting how many recognized words (tokens) there are for each language (in our case, Portuguese and English only) and selecting the language with more recognized tokens.

If we're unable to make this distinction with recognized tokens, we proceed as follows:

- For all languages, we check misspellings for unrecognized tokens with, at least, 3 letters.
- We obtain correct tokens from our dataset that are, at least, 60% similar with the unrecognized tokens (creating a list of suggested corrections).
- We select the language with more recognized tokens after spelling check, in other words, the language with more recognized and suggested tokens.

V. GRAMMAR CHECKER

One of the features that our chat bot has is the ability to check if the input given by the user is grammatically correct or not. There are multiple libraries in Python that, given one or more sentences, check grammar, spelling mistakes and return corrections for these sentences, like **Sapling** [8] or **LanguageTool** [5]. We experimented with these libraries, but came to the conclusion that, although very powerful and very complete, these libraries were slower than desired and some required either calls to external APIs or a server running locally. So, after some investigation, we decided to use **Context-Free Grammar**, since it is simpler to implement, while still providing a good control over what we consider grammatically correct.

Context-Free Grammars (CFG) are a set of recursive rules used to generate patterns of strings. These grammars have 4 kinds of components:

- **Terminal Symbols** - These are the characters that appear in the language/strings generated by the grammar. Terminal symbols never appear on the left-hand side of the production rule and are always on the right-hand side.
- **Non-Terminal Symbols** - These are placeholders for patterns of terminal symbols that can be generated by the non-terminal symbols and will always appear on the left-hand side of the production rules, although they can be included on the right-hand side. The strings that a CFG

produces will contain only symbols from the set of non terminal symbols.

- **Production Rules** - These are the rules for replacing non-terminal symbols. Production rules have the following form: variable \rightarrow string of terminals and non-terminal symbols.
- **Start Symbol** - This is a special non-terminal symbol that appears in the initial string generated by the grammar.

Usually, the terminal symbols in our grammar would be words like 'eat' or 'hello', which would then be associated with non-terminal symbols that would represent either the word or a category where that word fitted, like *verb* or *noun*. This way of constructing the CFG would make it cluttered and very extensive. To avoid this, and to focus only on the grammar rules, we utilized **taggers**.

Taggers take one or more characters/words and associate them with a **part of speech** category, like *verb*, *noun*, *adverb*, *etc.*. This means that, after the tokenization of the user input, we can put the tokens through the tagger and obtain the part of speech category that each token belongs to. This allowed for the terminal symbols in our grammar to be the part of speech categories, thus compacting the grammar and making it focused on the production rules.

In our chatbot, since it allows the user to use either **English** or **Portuguese**, we felt the need to utilize two different taggers, one for each language. For English, we used the tagger provided by **nlk**[7], which uses the *Penn Treebank*[6] tagset. Since the `nlk` tagger is not trained for Portuguese, we used a tagger developed and trained by *inoueMashuu* using the **Mac Morpho**[2] tagset, which can be found on his **Github**[4]. Although both taggers are not completely accurate, we felt that the advantage of not having to define every word in the grammar outweighed this disadvantage.

We can see an example of how the english tagger works in fig 4. In the example, the sentence 'Tell me a joke' was tokenized and then inputted in the tagger, which then returned for each word the part of speech category that the word belongs to.

The Portuguese tagger, as we can see an example of how it works in fig 5, works similarly to the English tagger, with the only difference being that the categories of each word come in Portuguese.

The English grammar can be seen in 12. This grammar was based on the **12 Grammars**[3] slides provided by the University of Colorado, which we then changed and improved to be in compliance with our tagger and to accept the kind of sentences that the bot would receive. As for the Portuguese grammar, we did not find any relevant papers or examples, so our Portuguese grammar, which can be seen in 13, was entirely developed by us through some common notions that we have regarding Portuguese, as well as trial and error during testing. It is worth to say that since both these grammars were either totally or partially developed by us in a short amount of time, they are incomplete and can have mistakes, which leads to some sentences that are grammatically correct to not be considered correct by our grammar and the opposite, meaning

VI. MODEL

One of the most important components in any robot is the way it will think and respond to any type of event made by an user, or any change in environment.


As the basis for the chatbot's mind, we decided to use one Machine Learning model, as well as a Vectorizer.

A. TF-IDF Vectorizer

For the Vectorizer, we decided to use a TF-IDF Vectorizer [10], which converts a collection of raw documents to a matrix of TF-IDF features.

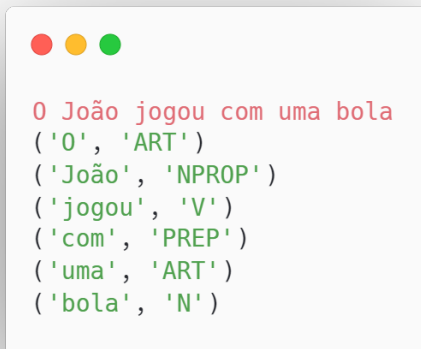
TF-IDF (Term Frequency - Inverse Document Frequency) is an algorithm that uses the frequency of words to determine how relevant those words are to a given document. It's a simple but intuitive approach to weighting words, allowing it to act as a great basis to our training data.

As an example let's look at a simple sentence, "tell me a joke", and the TF-IDF matrix in the Figure 6:



```
Tell me a joke
('Tell', 'VERB')
('me', 'PRON')
('a', 'DET')
('joke', 'NOUN')
```

Fig. 4. Example of the English tagger

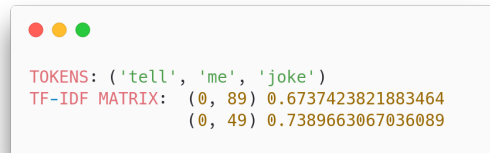


```
0 João jogou com uma bola
('0', 'ART')
('João', 'NPROPN')
('jogou', 'V')
('com', 'PREP')
('uma', 'ART')
('bola', 'N')
```

Fig. 5. Example of the portuguese tagger

that some sentences that are not grammatically correct pass are considered correct.

One feature made possible due to the way we are checking the grammar of the user input is displaying the syntactic tree of the last input the user gave. As we can see in figure 11, the user inputted 'what time is it', and, after the bot responded, the user asked to see the syntactic tree, to which the bot drew the syntactic tree of the sentence 'what time is it'. The tree shows that the sentence contains a **noun-phrase** comprised of a pronoun 'what' and a noun 'time' and contains a **verb-phrase** comprised of a verb 'is' and a pronoun 'it'.

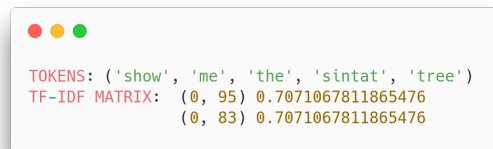


```
TOKENS: ('tell', 'me', 'joke')
TF-IDF MATRIX: (0, 89) 0.6737423821883464
                (0, 49) 0.7389663067036089
```

Fig. 6. TF-IDF matrix for the sentence "tell me a joke"

We can see that the matrix only has two entries, as the Vectorizer only categorizes the words "tell" and "joke" as important. Each word has different importance values, which can tell us that the word "joke" is more important than the word "tell".

Now let's look at a more complex sentence, in the Figure 7:



```
TOKENS: ('show', 'me', 'the', 'sintat', 'tree')
TF-IDF MATRIX: (0, 95) 0.7071067811865476
                (0, 83) 0.7071067811865476
```

Fig. 7. TF-IDF matrix for the sentence "show me the syntactic tree"

The TF-IDF Vectorizer only recognizes two words as important, which are the tokens "sintat" and "tree", which consequently will give us the syntactic tree of the last sentence.

B. MLP Classifier

After creating the training data, we used a Multi-layer Perceptron Classifier [11], which optimizes the log-loss function using LBFGS or stochastic gradient descent.

In our case, after testing different activation functions and different solvers, using different learning rates and even maximum number of iterations, we came to a conclusion that using 'identity' (a no-op activation function useful to implement linear bottleneck, which limits $f(x) = x$) was the best choice, while also using an optimizer in the family of quasi-Newton methods, 'lbfgs' (recommended to use in smaller datasets, such as ours). If we ever increase the number of intents by a lot, we should change the solver to 'adam', which takes more time to converge, but works pretty well on training and testing large datasets.

We also decided to use 3 hidden layers, with 8 nodes each, and go for 3000 max iterations, instead of minimum convergence.

To predict the response to a message, we simply need to tokenize the message, transform the tokens into an importance vector, using the TF-IDF Vectorizer, and predict the intent.

We also added a new condition, which returns a set intent, "Anything", in case the model could not predict any intent based on the message. The chatbot will then return a message: "I didn't understand... I'm sorry...", or in portuguese: "Não percebi... Desculpa...".

After any intent is predicted, the chatbot will select any random response from the selected intent, and print it to the terminal for the user to see.

VII. KNOWLEDGE ACCUMULATION

Our bot is capable of remembering information given by the user at any time of the conversation, and remember when asked for it. For this, we took two different approaches so that we could cover both English and Portuguese languages, and we opted to use only the last one.

A. SpaCy [12] Approach

We use the spaCy package to gather possible entities in the prompt given by the user. We use a file with a collection of words with a tag associated for the English language, and then train a spaCy pipeline with that file. Finally we use this pipeline to recognize and extract the entities from the user input: they will come in a bi-dimensional array, where each array stores the entity recognized in the input, and the tag associated to that word. For example, India would be coupled with COUNTRY, and John with PERSON.

Sadly, there are no good entity training files with the Portuguese language, and some of the information that we want to retain isn't recognized in the input phrase. For example, if the user says "i like pineapple" the word pineapple, which is supposed to be registered as an LIKE, will not be recognized by the entity pipeline. Because of these reasons, we decided not to use this approach and took a different approach in order to successfully gather all the information that we intend to save.

B. Dataset Approach

To solve the previous problems, we modified the dataset so that it would save the name of the entity that appears in this intent. The dataset has an entry called entities, where it states what entities are used in this intent, for example PEOPLE. The answers, which correspond to the prompt of the user, have the tag ;NULL_i in the position of the entity, so that the entity provided in the input is replaced (with the tag ;NULL_i) and recognized by the bot with the corresponding intent. The responses, which have the tag with the corresponding entity needed, will be fetched from a dictionary that has all the entities gathered until the moment.

When a token isn't recognized and it is located at the end of the phrase, we recognize it as an entity, because all the prompts in the database locate the entity in the end of the phrase. This guarantees the recognition of the entities independently of the phrase.

C. Other improvements

We also added the possibility of creating new chatbot responses when the user finds a response given by the bot to be not fulfilling. The bot will remember that response and use it in the future.

VIII. 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 model as it currently is the bottleneck of the bot
- Improve the Vectorizer
- 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 tokens from our dataset

IX. CONCLUSION

In conclusion, we can say that, even though the chatbot isn't fully optimized, the results are very satisfactory. In a little time, we created a complex chatbot with a big variety of features, using machine and deep learning models, and other tools, which were all very new to us.

We also completed the tasks set for the chatbot. Now anyone can have various conversations with our chatbot.

X. APPENDIX


The following figures are various conversations made between a user and the chatbot, and what the chatbot can actually do:


```
$ python chatbot.py
Chatty: Hello, I am Chatty. How can I help you ?
    Olá, eu sou o Chatty. Como posso ajudar ?
> Olá
Chatty: Olá humano, por favor indique o seu Chatty username
> Eu sou o filipe
Chatty: OK! Olá filipe, como posso te ajudar?
> Está tudo bem?
Chatty: Olá, eu estou ótimo, e tu?
> Estou bem
Chatty: Ótimo! Ei filipe, como posso te ajudar?
> Eu amo bananas
Chatty: Olá humano, por favor indique o seu Chatty username
> Eu nao disse isso
Chatty: Desculpa, o que seria uma boa resposta a: Eu amo bananas
> Eu tambem gosto muito de bananas
Chatty: Thank you for your help
    Obrigado pela ajuda !!
> Como me chamo
Chatty: filipe, como posso ajudar?
> Sabes alguma piada
Chatty: O condenado à morte esperava a hora da execução, quando chegou o padre:
    Meu filho, vim trazer a palavra de Deus para você.
Perda de tempo, seu padre. Daqui à pouco vou falar com Ele, pessoalmente. Algum recado?
> Eu tambem gosto muito de bananas
Chatty: exit
Chatty: Goodbye !
    Adeus !

$ python chatbot.py
Chatty: Hello, I am Chatty. How can I help you ?
    Olá, eu sou o Chatty. Como posso ajudar ?
> Eu amo bananas
Chatty: Eu tambem gosto muito de bananas
> Compreendes o que estou a dizer?
Chatty: Compreendo
> mostra me a árvore sintática
Chatty: Aqui está a árvore semântica:Compreendes o que estou a dizer?

      S
     / \
    /   \
   /     \
  /       \
 /         \
NP         VP
|          / \
N         /   \
Compreendes  V  VP
              / \
              /   \
             /     \
            /       \
           /         \
          /           \
         /             \
        /               \
       /                 \
      /                   \
     /                     \
    /                       \
   /                         \
  /                           \
 /                             \
V                             VP
|                             / \
V                         /   \
V                         /     \
V                         /       \
                        /         \
                       /           \
                      /             \
                     /               \
                    /                 \
                   /                   \
                  /                     \
                 /                       \
                /                         \
               /                           \
              /                             \
             /                               \
            /                                 \
           /                                   \
          /                                     \
         /                                       \
        /                                         \
       /                                         \
      /                                           \
     /                                             \
    /                                               \
   /                                                 \
  /                                                   \
 /                                                     \
/                                                       \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /               \
        /                 \
       /                   \
      /                     \
     /                       \
    /                         \
   /                           \
  /                             \
 /                               \
/                                 \
N               V  VP
|               / \
Compreendes  /   \
              /     \
             /       \
            /         \
           /           \
          /             \
         /              
```

Fig. 8. Appendix - Portuguese

A terminal window with a black background and white text. At the top left, there are three colored circles: red, yellow, and green. The terminal shows a command prompt where the user has run 'python chatbot.py'. The chatbot responds with a greeting and offers help. The user then asks for help with bananas, and the chatbot responds with a list of banana-related items. The user then says 'bye' and the chatbot responds with a farewell message. Finally, the user types 'exit' and the chatbot says 'Goodbye !'. The user then types 'Adeus !' which is not followed by a response from the chatbot.

```
$ python chatbot.py

Chatty: Hello, I am Chatty. How can I help you ?
    Olá, eu sou o Chatty. Como posso ajudar ?
> I love bananas
Chatty: I also like bananas
> What do you love ?
Chatty: I also like bananas
> bye
Chatty: Bye! Come back again soon.
> exit
Chatty: Goodbye !
    Adeus !
```

Fig. 9. Appendix - Entities

REFERENCES

- [1] *aliam. simple_chatbot*. URL: <https://www.kaggle.com/code/aminianam/simple-chatbot/input>.
- [2] *CONJUNTO DE ETIQUETAS (TAGSET) do MacMorpho*. URL: <http://nilc.icmc.usp.br/macmorpho/macmorpho-manual.pdf>.
- [3] *Context-Free Grammars for English*. URL: <http://www.cs.uccs.edu/~jkalita/work/cs589/2010/12Grammars.pdf>.
- [4] *inoueMashuu. Conjunto de POS-taggers treinados para classificação gramatical de frases em português*. URL: <https://github.com/inoueMashuu/POS-tagger-portuguese-nltk>.
- [5] *LanguageTool. LanguageTool is a multilingual grammar, style, and spell checker*. URL: <https://languagetool.org/>.

```

$ python chatbot.py

Chatty: Hello, I am Chatty. How can I help you ?
Olá, eu sou o Chatty. Como posso ajudar ?
> hello
Chatty: Hi human, please tell me your Chatty username
> my user is filipe
Chatty: OK! hi filipe, what can I do for you?
> what time is it
Chatty: The time is 14:41
> Show me the syntactic tree
Chatty: Here is the syntactic tree from the last sentence: what time is it

      S
     / \
    NP  VP
   /  \ /  \
  PP  NP V  NP
 /  \ /  \ /  \
P   N V   P   N
what time is it

> clever
Chatty: Thanks, I was trained that way
> clever
Chatty: [Language not detected]
Chatty: Thank you, I was trained that way
> clever
Chatty: [Language not detected]
[0] Did you mean: clever
[1] Did you mean: clever
>>> 0
Chatty: I was trained that way
> bye
Chatty: I didn't understand... I'm sorry...
> bye
Chatty: Bye! Come back again soon.
> exit
Chatty: Goodbye !
Adeus !

```

Fig. 10. Appendix - English

```
$ python chatbot.py

Chatty: Hello, I am Chatty. How can I help you ?
    Olá, eu sou o Chatty. Como posso ajudar ?
> I love bananas
Chatty: I also like bananas
> love bananas I
Chatty: You should check your grammar!
    Deverias verificar a tua gramática!
> bye
Chatty: Bye! Come back again soon.
> exit
Chatty: Goodbye !
    Adeus !
```

Fig. 11. Appendix - Grammar Example

- [6] Mary Ann Marcinkiewicz Mitchell P. Marcus Beatrice Santorini. *Building a large annotated corpus of English: the Penn Treebank*. URL: <https://catalog.ldc.upenn.edu/docs/LDC95T7/c193.html>.
- [7] nltk. *nltk.tag package*. URL: <https://www.nltk.org/api/nltk.tag.html>.
- [8] Sapling. *Language model copilot for customer-facing teams. Respond twice as fast*. URL: <https://sapling.ai/>.
- [9] Saumyab271. *Stemming vs Lemmatization in NLP: Must-Know Differences*. URL: <https://www.analyticsvidhya.com/blog/2022/06/stemming-vs-lemmatization-in-nlp-must-know-differences/>.
- [10] scikit-learn. *sklearn.feature_extraction.text.TfidfVectorizer*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.
- [11] scikit-learn. *sklearn.neural_network.MLPClassifier*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html.
- [12] spaCy. *Industrial-Strength Natural Language Processing in Python*. URL: <https://spacy.io/>.

```

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

```

Fig. 12. Appendix - English Grammar

```

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

```

Fig. 13. Appendix - Portuguese Grammar

- [13] *Stemmer Documentation*. URL: <https://www.nltk.org/howto/stem.html>.
- [14] *WordNetLemmatizer Documentation*. URL: <https://www.nltk.org/api/nltk.stem.WordNetLemmatizer.html?highlight=lemmatization>.