# Retrieval-based Chatbot
## Natural Language

**Group 34**
Margarida Campos 77039
Gonçalo Faria 91073
Gustavo Morais 97152

## 1 Introduction

The goal of this project was to build a system for *Balcão do Empreendedor* that can answer user questions based on:

- Set of questions with respective answer from Frequently Asked Questions(FAQs)
- Similarity between user's input and set of questions

Essentially we wanted to build a retrieval-based chatbot in the sense that it will return one of predefined set of answers and will select that question based on a ranking. In this case, the ranking will be defined by the distance (inverse of similarity) between questions.

## 2 Proposal

We started by designing a system that could easily integrate different tests and hypothesis. With the general workflow, represented in *Figure 1*, we set on to find an architecture that allowed us to play with processes,represented in *gray background*, and iterate over the workflow to get the best evaluation score.

We dealt with the implementation of the architecture and software first so we could then test every hypothesis fast thus focusing our attention and efforts in the actual analysis and exploration.

[1] **Some Definitions:**

- Preprocessing options - dictionary with all preprocessing possibilities as keys and possible values of 0 and 1
- List of metrics - list of metrics chosen to be tested
- Configuration - pair of {Preprocessing options,List of Metrics}

In this way we could run tests for all our hypothesis and store both results and accuracy so we wouldn't lose track of our work and also deep dive on the "why" of the results later.
An example of a configuration would be:

```
[{lowerCase: 1,
tokenize: 1,
stem: 0,
removePuntuation: 1,
tokenize: 1,
tfIdf: 0,
tokenCount: 0,
tokenize: 1,
removeNumbers: 0,
stopwordsNltk: 1,
stopwordsMinimal: 0,
stopwordsAnalysis: 0},
'jaccard',
'editDistance'
]
```

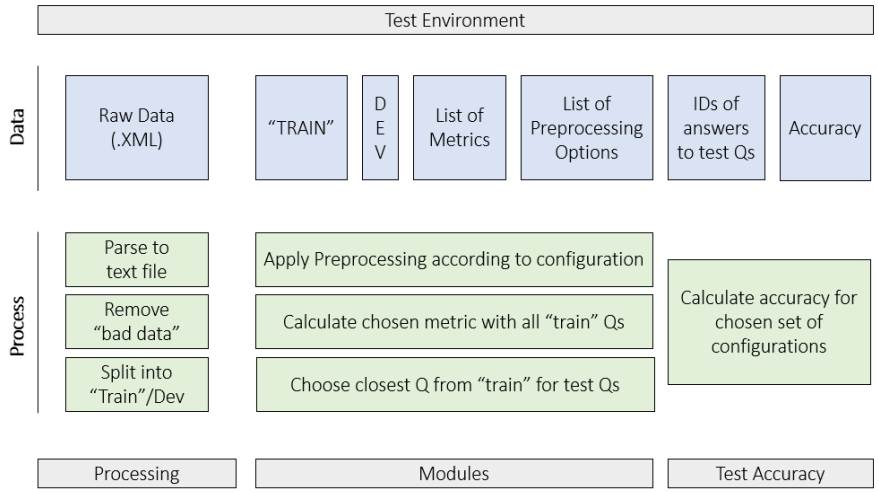


Figure 1: Implemented workflow of testing

[1] **Q** in this document always stands for question

Apart from the main idea of choosing a metric and returning the answer corresponding to the question that minimized the distance to the input, we also wanted to try a different approach:

1. Evaluate the distance between input Q and all other questions
2. Calculate the mean between measures of the same group,i.e. with the same answer ID
3. Assign ID with the best average measure

The thought behind this was that only the answer we give the user is actually relevant, and by evaluating the input with the entire group we would be able to avoid situations like:

$Q_i$: 'Será que seria possível eu adquirir a minha certidão no Registo Civil?'

$Q_{id=x}$: 'Será que seria possível eu adquirir a minha carta no Registo Civil?'

$Q_{id=x}$: 'Onde posso fazer a minha certidão?'

A significant number of metrics would return a high similarity between the first two questions, whilst it's clear for us, humans, that the questions refer to different things. On the other hand, by averaging the result with the third question (same id as second) it would return a smaller similarity, thus diminishing the impact that the same formulation of one specific sentence would have had. By averaging the results of the metrics we could get a sense of similarity with the entire group.

# 3 Corpora

From now on, the following definitions apply:

- Corpus: the set of all questions text
- Document: each question's text
- Term: a word
- Vocabulary: the union of all terms that occur in Corpus
- Question Set: group of questions with the same ID

The Corpus was built by every group of students, who were tasked with paraphrasing the questions in the FAQ list, creating the question sets. The goal is to facilitate a match between the question introduced by the user and one of the FAQs. It also comes in handy because it allows us to use one paraphrase from each question set to create the test set. The rest of the questions form the training set. However, this can also inflate our accuracy, since the questions within a question set are usually quite similar between themselves. Figure 2 illustrates this similarity among questions in the same question set:

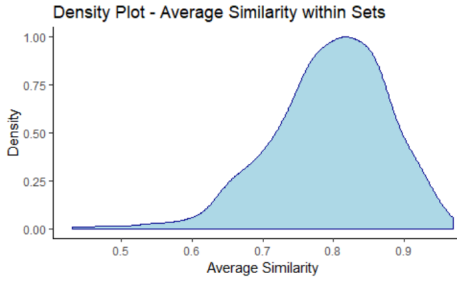


Figure 2: Density distribution of similarity in set

Our Corpus consists of **2641** questions labeled with **615** distinct IDs. Over **90%** of the IDs have **4** questions associated with an average of **4.28 questions per ID**. There are a few outliers with over 20 questions. Our vocabulary has 1983 terms. Analyzing the Corpus we found three main problems:

- Empty questions - 6 questions were " "
  - We removed these questions
- Duplicate Questions sets - Same questions with different answer IDs
  - We removed the duplicate sets from train data
- Duplicate answers - different sets of questions (different IDs) with the same answer
  - We moved the questions from one set to the other, so they would all have the same answer Id.

We removed the empty questions and the duplicates because there was no extra information in those.
Then we set out to understand which words occurred more often has to not only understand the Corpus but to find suitable (and personalized) lists of **stopwords**.


e

Figure 3: Original Corpus wordcloud


Figure 4: After Stopwords removal wordcloud

# 4 Results

We used as **baseline** the model that used Jaccard Distance as measure with Tokenizing as the only Preprocessing technique. The accuracy on the test set was **80.63%**. We then tested with the following Preprocessing techniques and measures:
**Preprocessing**:

1. Remove punctuation
2. Remove numbers
3. Lowercase
4. Remove diacritics
5. Remove stopwords (different lists were tested)
6. Stemming
7. Embedding using tf-idf

**Measures**:

1. Minimum Edit Distance (MED)
2. Jaccard Distance
3. Dice
4. Cosine Distance
5. Euclidean Distance

We've summarized the accuracy results for some configurations in the heatmaps in *Figures 4* and *5*. Most of the results are were aligned with expectations. For instance it's intuitive that performance increases with the removal of punctuation and stopwords.
An example behind the intuition:

$Q_{35}$ : Quem pode pedir uma Certidão Permanente online?
$Q_{teste}$ Posso pedir certidão permanente online?

It is clear, for us humans, that the two questions have the same answer. In an ideal world the distance would be 0

| Preprocessing | Jaccard Distance | $\cap$ | $\cup$ |
|---|---|---|---|
| None | 0.73 | 3 | 11 |
| - Punctuation | 0.8 | 2 | 10 |
| + Lowercase | 0.5 | 4 | 8 |
| - Stopwords | 0 | 4 | 4 |

We can see that by adding preprocessing techniques we could extract the essential meaning of each sentence and decrease distance to zero. It's important to notice that although removing punctuation didn't help in this example, it improved model's accuracy as it helps us deal with more complex scenariol Jurafsky, James H. Martin s with, for example, commas in the middle of the sentences.

We concluded that the best results were obtained using Cosine and Euclidean Distance measures applied to TF-IDF vectors.
This result, however, should be taken with a grain of salt as the execution time is substantially larger than some configurations.

| Method | Time to test($s$) |
|---|---|
| Preproccess + Dice | 3.27 |
| Preproccess + Jaccard | 3.97 |
| Preprocess + MED | 47.14 |
| Prep. + TF-IDF + Cosine | 51.42 |

As to the accuracy, however, the cosine and euclidean distances showed better results. The Figures below show the accuracy of each measure when applied to the test set. The results shown were obtained with the method of using the average similarity between the test question and each question set (instead of with each individual question). Results with the simple method are similar but all Configurations obtained slightly lower accuracy.

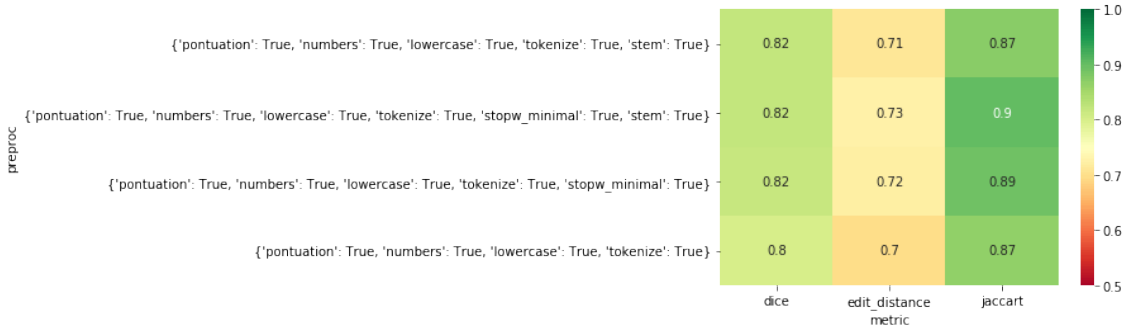


Figure 5: Accuracy results-without TF-IDF using avg. measure in group


Figure 6: Accuracy results - with TF-IDF using avg. measure in group

The best results were obtained with the Average Euclidean distance within ID with all the Preprocessing techniques except the removal of stopwords. This may seem surprising, but our understanding is that TF-IDF essentially plays the role of the stopwords removal by giving the value 0 to insignificant words (which does an even better job).

**Threshold**:When deciding on the threshold value, we thought we could either use the maximum distance between a correctly marked test set question and its respective corpus question, or the minimum distance between an incorrectly marked test set question and its respective corpus question. We decided that if the distance of $Q_{input}$ to other questions was over **1.1** the system would not return an answer - meaning it will return $ID = 0$. This number was chosen because it was near the maximum value of distance in between correctly marked test set questions and the training corpus and it also excluded most questions we tested that had nothing to do with the problem's context.

# 5 Conclusions

The main obstacle we came across was the actual cleaning of the Corpus. We were obtaining very low accuracy values because we had not dealt properly with the duplicate question sets and answers. We think the major lesson learned was to actually look at the data and continuously reflect on the results and try to make sense of the reasons for them.
If we had more time we would probably move ahead to:

- Lemmatization
- Improve performance of TF-IDF to make the bot faster
- Tagging different FAQs using Classification

# References

[1] Daniel Jurafsky *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall 2009.

[2] NLTK Documentation
https://www.nltk.org/