

Automatic Summarization System

Christian Faccio¹ *, Elena Lorite¹ †, Rebeca Piñol Galera¹ ‡ and Paula Frías Arroyo¹ §

¹University of Alicante

Abstract

In this project we developed an automatic summarization system, which is able to effectively summarize the information present on a set of documents. The process is divided in two steps: information retrieval, done using a non-neural approach with RegEX and a neural approach with transformers, and generation, completed with an encoder-decoder model. We finally analyzed the effectiveness of both approaches, showing that modern state-of-the-art transformers are able to capture more information than standard information retrieval methods, but at the cost of efficiency and possible problems like hallucinations.

Introduction

Summarizing the text contained in a document is a difficult task even for people. It necessitates the individuation of the most important topics, then the selection of the most important part of them and finally the generation of a smaller text starting from that. **Automatic Summarization Systems** are employed to help in this task: they should be able to summarize the text of the documents autonomously or with limited and small human help like in this case. It is a task which divides in two parts: the first is the individuation and extraction of the most important information, while the second is the generation of new text starting from that. In our case, the information to be extracted is defined beforehand by us, so the first architecture just needs to find the most relevant parts of the text with respect to that. The developed code and relative data can be found in our GitHub repository.

Data

The documents used in this project are scholarships for students published in the Boletín Oficial del Estado (BOE) and can be found in the docs/ folder. They are 5 documents in total and are mostly similar with respect to the format.

The information extracted is detailed and specific:

- academic year;
- total budget;
- income thresholds;
- eligible programs;
- scholarship components;
- application period;

- academic requirements.

We believe that this data can represent most of the useful information contained in the documents and a special consideration must be done regarding the type of data extracted, since it is not just common text but comprises ranges, dates, prices and structured information. Given this variety of data types, the information retrieval part had to be specifically tailored for this task, as we will describe in the next section. Finally, the extracted information has been written on a JSON file in a structured manner, which can be found in the output/ folder.

Architecture

We divided the whole architecture in two components: the *information retrieval (IR)* part and the *generation* part.

Information Retrieval

This was the part in which we spent most time on. Without correct and reliable data, the summarization cannot be done, so this is probably the most important part of the system. There are different methods to be used here, but the most common are the use of *regular expressions*, which are not flexible at all but can extract information with special format like dates, the use of *similarity search* between a query and the chunks of the text, which is more flexible but contains hidden difficulties like the choice of the chunking method, and finally the use of a *transformer* model, which is the state of the art and is flexible, even if costly and complex.

Even if usually a transformer architecture is the best choice, it may not be for every situation. In fact, the best trade-off between performance and efficiency depends on the characteristics of the problem at hand. In this case, chunking the text and then use similarity search between a query and them was not

*christianfaccio@outlook.it

†elenalorite@gmail.com

‡rpg80@alu.ua.es

§pfa13@alu.ua.es

the best approach. We, in fact, tried to use a semantic chunker to split parts of the text which are coherent with respect to a speech topic, saved them in a vector store and ran similarity search with the topics of interest. The results were good, the data extracted were coherent with the query, but the main problem was that this way was impossible to extract detailed and structured information as we needed to. This necessity comes from the fact that the documents at hand contain information regarding scholarships, so a summary of them should include detailed information which are helpful for a possible candidate, and that include dates, ranges and mostly numbers. Thus, a better approach in this specific case is the use of regular expressions.

We would like to emphasize that this method was possible just because the documents are similar in writing and format, thus the lack of flexibility of the method. But, given that we needed structured and specially formatted information, this was a good approach. The results are shown in the `info.json` file and are pretty good indeed, even if in some cases we were not able to extract all the data. Unfortunately, with small text variations this method stops working correctly.

We also used the approach with a transformer, to see and compare the performance with the former method. It is much more flexible and let us obtain better results with less manual and strict code, even with the cost of higher complexity and with some hallucination. The information for this approach have been saved, always with a structured manner, in the `info_mistral.json` file.

Generation

Evaluation

Conclusions