# Matching Patient Cases to Clinical Trials

James Furtado[61177], Ricardo Gonçalo[60519], and Diogo Silva[53058]

NOVA School of Science and Technology | FCT NOVA
https://www.di.fct.unl.pt/

## 1 Introduction

In the context of Information Retrieval course, lectured at Nova School of Science and Technology edition 2023/24, we were proposed to develop an information retrieval system that given the description of a patient, in natural language, will find the best clinical trials the patient is fit for.

For the first phase of the project, we were assigned to finish the implementation of two retrival models we were taught during the lectures: a space vector model called TF-IDF[1] and a probabilistic model[2] that uses a Unigram language model with Jelineck-Mercer (JM) smoothing.

Apart from the implementation we were challenged to find good ways we can measure the performance of each model. Our implement some of the sujections of the professor and tried to come up with a few as well.

The following sections of this report will dive into the methodologies employed, and the discussion of the results.

## 2 Implementation & Methodology

Regarding the implementation there isn't much to discuss about, since the it was pretty much straight forward. The implementation of the models can be found in file **index.py** on the zip file delivered with this document. The remaining **.py** files implements some useful functions that are used in the **project.ipynb** file. The latter is the main file of the project and aside from calls to the models it has some code to generated charts that will be useful to measure the performance of the models, thorough the rest of the project.

Regarding the methodology, the only model that had a significant room for change was the LMJM, because of the parameter $\lambda$, part of JM smoothing. In order to choose such parameter, we basically tested the whole set of queries provided with different values for $\lambda$. From the results we built figure 1. The chosen value, as shown by the vertical red line, was *0.2*, which as can bee seem scores fairly very well on every metric and looking at the neighbouring it seems to be stable, thus making it a great choice.

As recommended by the professor, on this first phase of the project, we only used the **brief_title** field from the clinical trials and the **summary** version of

[1] Which stands for Term Frequency-Inverse Document Frequency

[2] Refered as LMJM through the rest of the document.

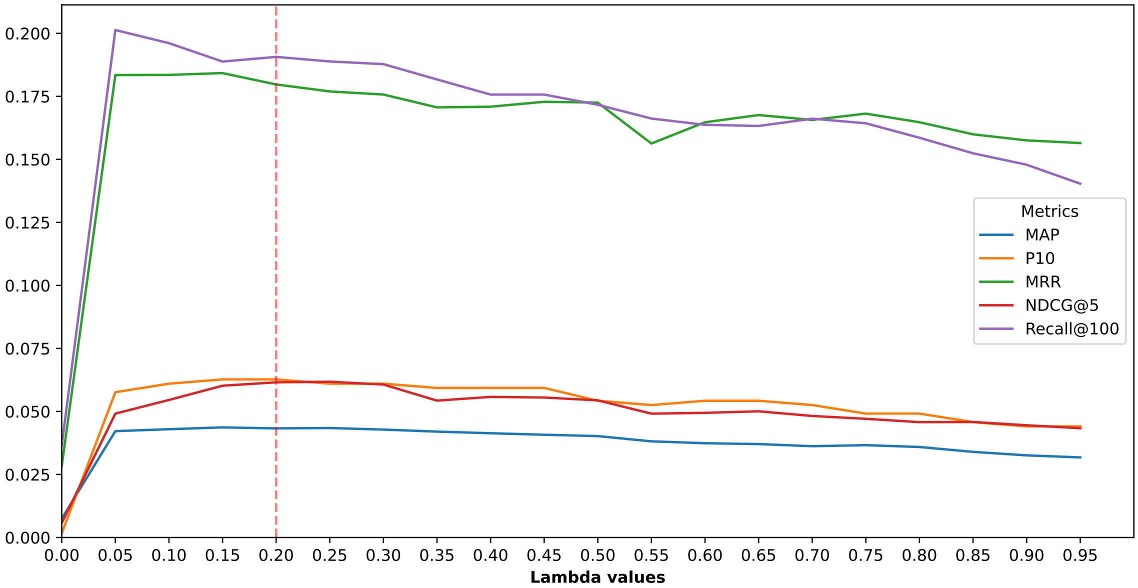

Fig. 1: LMJM performance for different values of $\lambda$

the queries. Since they are smaller and faster to process and we are still in the early stages of the project. This was the case for the methodology to choose $\lambda$ and for the experiments.

## 3 Experimental Setup

As mentioned in the section above, at this point there isn't much we can do to improve our solution so the experiments were also straight forward. We basically ran the whole set of queries in the both models and from the results we built some charts. These charts gives us some insight on how well the models are performing and how they compare to each other. The charts are presented and discussed in the next section.

## 4 Results Discussion

The first chart we present is in the figure 2, where can be seem the performance of each one of the models in 4 different metrics. Some of these metrics were discussed during the lectures and the others are presented in the bibliography of the course. It's important to notice that each one of them varies between 0 and 1, where 1 is the best possible score.

As can bee seem the TF-IDF model scores slightly better in all 4 metrics, which is not surprising since as stated in the section 2 we used a summarized

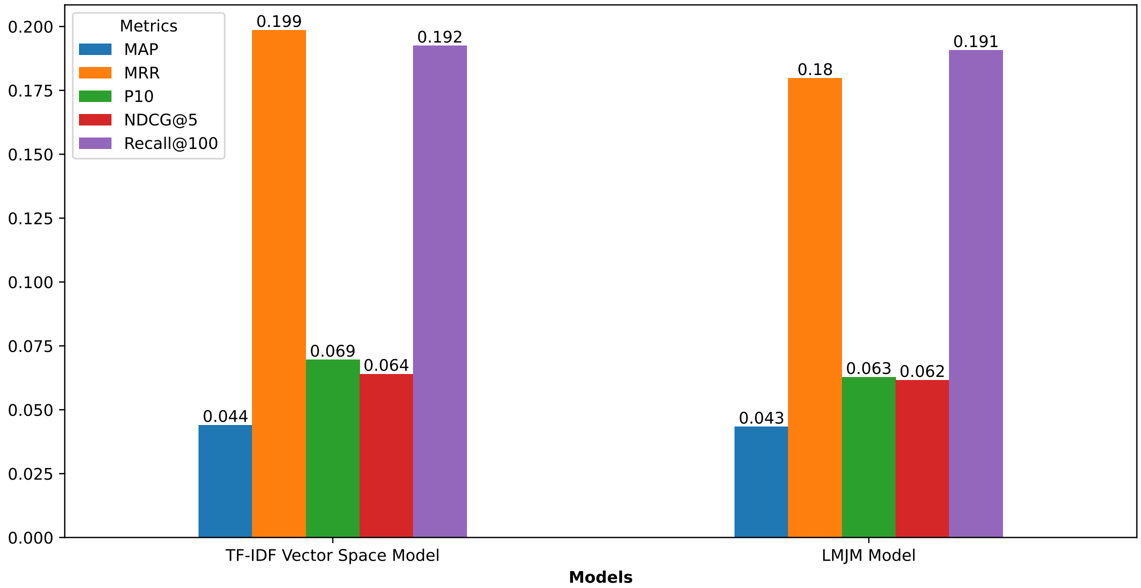

Fig. 2: Summary of the overall performance of the models

version of the documents and the queries. Since TF-IDF really cares about words presented in the query and the document its great for it since summaries usually to have only the most important words, thus being a good fit for TF-IDF. As for LMJM it works with a Language Models that tries to approximate the probability of the query being generated by the document, using summaries is not good, because the probabilities will be very biased, because there is a very small sample of words being considered.

The next chart we want to present is in the figure 3. It displays the precision recall curve of each one of the models along with their respective confidence interval. As it can be seem, once again TF-IDF model does slightly better than the LMJM model. What can be taken is that the TF-IDF model is slightly more precise than the LMJM model, because the relevant documents are ranked closer top than in LMJM model. Also, taking in account what was considered during the lectures, it seems that the relevant documents appears in the rankings one after another, because of the shape of the curves. This is good since if the irrelevant on top are dealt with the overall performance of the models will increase. One last comment we want to make about the chart is that as recall increases the precision starts to of the models converge to the same value, which is totally expected because in the limit (when recall is equals to 1) the precision is the same for whatever model is used for this project.

There are other charts we built and we would like to present, but because of the pages constraints we had to discard them. They can still be found in the

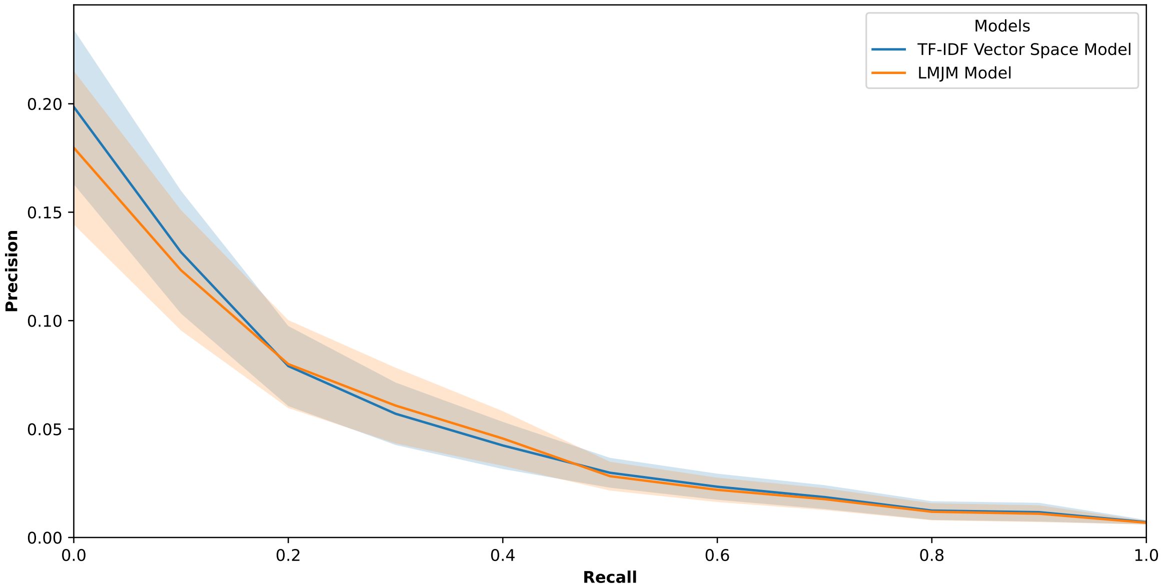

Fig. 3: Comparison of the Precision Recall Curves of the Models

**project.ipynb** file. The most important thing to be taken out of this results is that TF-IDF model performed better than LMJM and we made clear the reason why this is expected. But if we really look through the results we can see that the overall performance of the models is not great. The metrics used they vary between 0 and 1, but obtained results we are still in the lower half. We believe that is because of the following: (1) we used summarized version of the queries and the documents, which may be good for TF-IDF but still it can hide important words that are crucial to the discrimination of the documents. (2) The rankings obtained are blind in the sense that they don't consider the age and gender constraints of the patients. (3) Because we are not taking advantages of other features like name entities and others. We believe that as the course moves on we will learn more about this and other techniques that will help us improve the performance of the models.

## 5 Conclusions

We implemented two retrieval models, named TF-IDF and LMJM and we tested the performance of each one of them. TF-IDF did slightly better but in the end the results aren't great. We believe that as the course moves on we will learn more about how to improves the results of the models, which will give us run for more experiments and hopefully better results.