Matching Patient Cases to Clinical Trials Phase 2

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 this project phase, our objective is to assess the clinical trial relevance for individual patient cases by leveraging various sections within the clinical trial document as predictive features. To achieve this, we will employ Vector Space Model and LMJM models to compute predictor signals. The integration of these signals will be accomplished through the implementation of a learning to rank approach, specifically utilizing a linear regressor such as the logistic regression model. The ultimate goal is to train the model to adeptly combine the distinctive strengths of VSM and LMJM across different document sections, producing a comprehensive document relevance score for each patient case. This approach allows us to create an ensemble of models, harnessing the power of diverse algorithms to enhance the accuracy and effectiveness of clinical trial relevance predictions.

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

2 Implemention & Metadology

The implementation of the LETOR model 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, throught the rest of the project.

In the first section of our project we start by loading the queries and our initial VSM and LMJM models taking as input the different fields of the Documents (clinical trial), such as the brief_title, brief_summary, detailed_description and criteria, with a chosen lambda of 0.2.

After that, we calculate all metrics for all individual models and plot the results on a bar graph and on a precision-recall curve, which will be discussed later.

In the following section we generate the scores for all models and queries using our calc_ranking function to obtain the scores given a model and a query, and we store them in a dictionary with key being the name of the model and the value another dictionary with the query id as key and a DataFrame with all the scores as value of the dictionary.

We then select the LMJM(Detailed Description) as the best model to rank the queries and split them in test and train queries, which is achieved by focusing on creating train and test sets for queries, considering their precision at 10 values. The splitting is based on the order of query IDs sorted by their precision at 10.

The following function get_query_doc_scores is designed to retrieve scores for a specific query and document pair from a collection of document scores. It iterates over different models, extracts the relevant DataFrame for the given query, and retrieves the score associated with the specified document ID from each model.

The following functions are related to the training where the first function generates pairs of input features (xx) and binary labels (yy) based on relevance judgments for a given set of query IDs. It iterates through the specified queries, extracts relevant documents and their relevance labels, and constructs pairs using the scores obtained from the get_query_doc_scores function. The resulting feature vectors (xx) consist of scores for each query-document pair, while the labels (yy) are indicate whether the document is relevant or not. The second function is pretty much the same but we give relevance 0 to non-judged documents, and in the third function, related to the hard negatives strategy, we use the non judged and non relevant documents from the top 100 of our initial rank plus all the relevant documents of the rank.

Finally, we train three models to correct data imbalance. In the first one we use logistic regression to train the model with parameters C = 0.5 and then we fit the features and labels. In the second one we use the same weight per class. In the third one we add different values per class, and decided to weight the class 0 with value 1 and the class 1 with value 5, as the positive class is more important.

3 Quantitative Results

3.1 Comparison and discussion of all individual models across all metrics

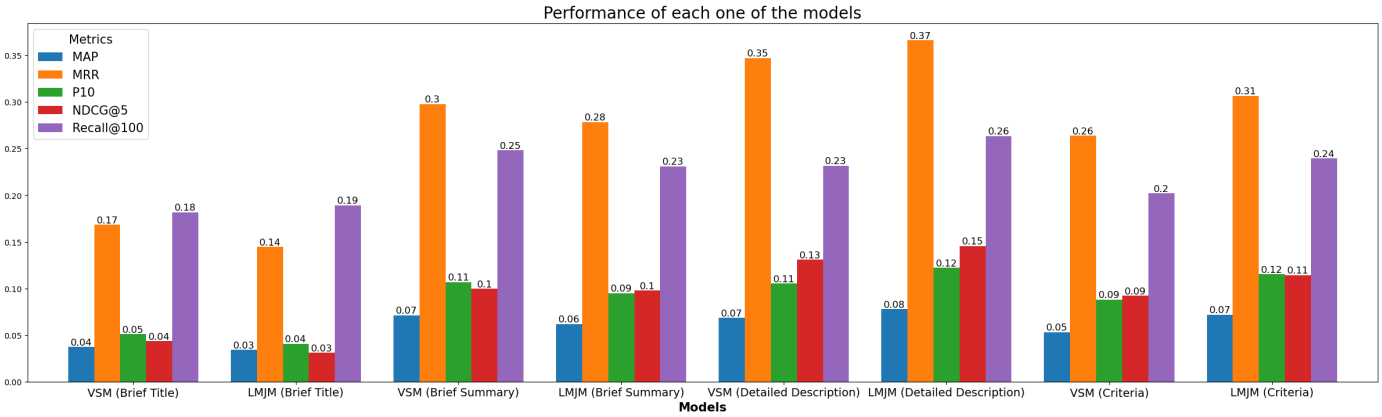

Fig. 1. Performance of each pair of type of model and corpus

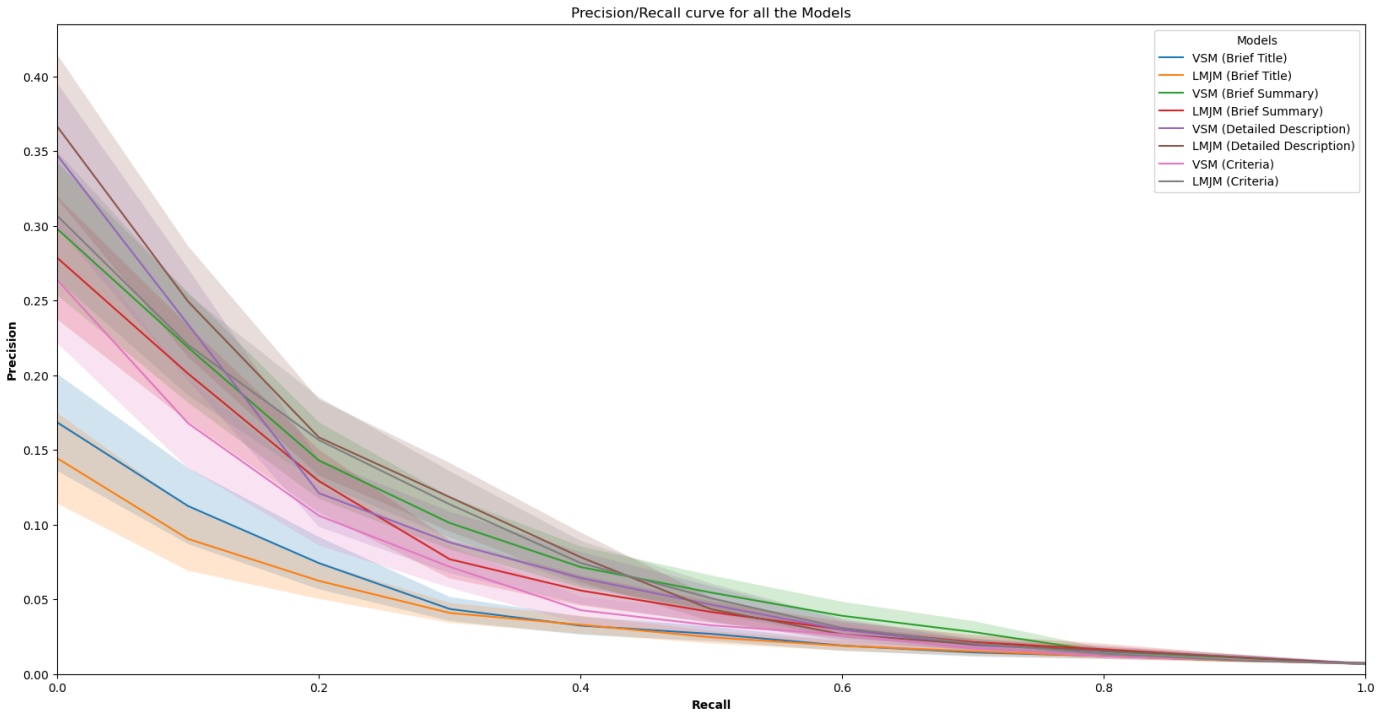

Fig. 2. Precision and recall for different models

We know that the goals of the system are recall-oriented, and from analysing the figures above, we get the best results with the recall and mrr metrics since the mrr metric accounts for both the precision and the rank of relevant documents and the recall metric focuses on the proportion of relevant documents retrieved within the top 100 ranks.

From the graphics we conclude that all individual models perform fairly well, except the first two (VSM(Brief Title) and LMJM(Brief Title)) that use the Brief Title field and that the ones that perform the best are the models that use the Description field, followed by the models that have the criteria as input.

One last comment we want to make about the chart is that as recall increases the precision starts to 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.

3.2 Comparison and discussion of the letor model across all metrics

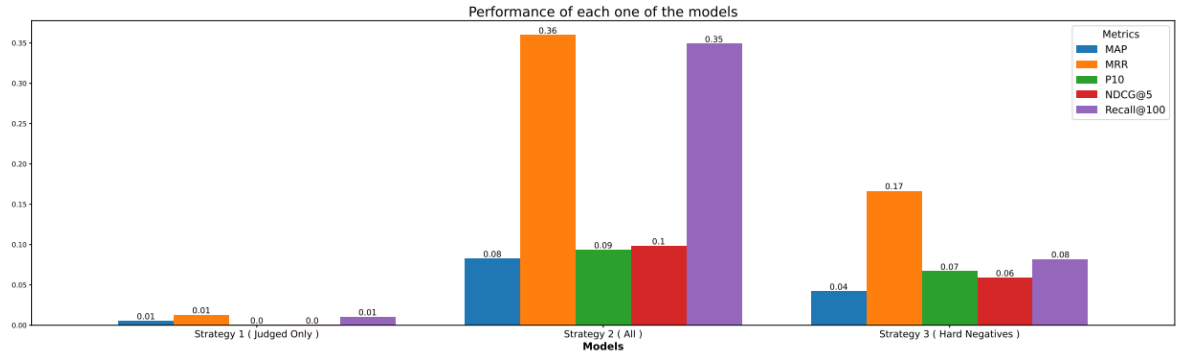

Fig. 3. Performance of generated model for each strategy

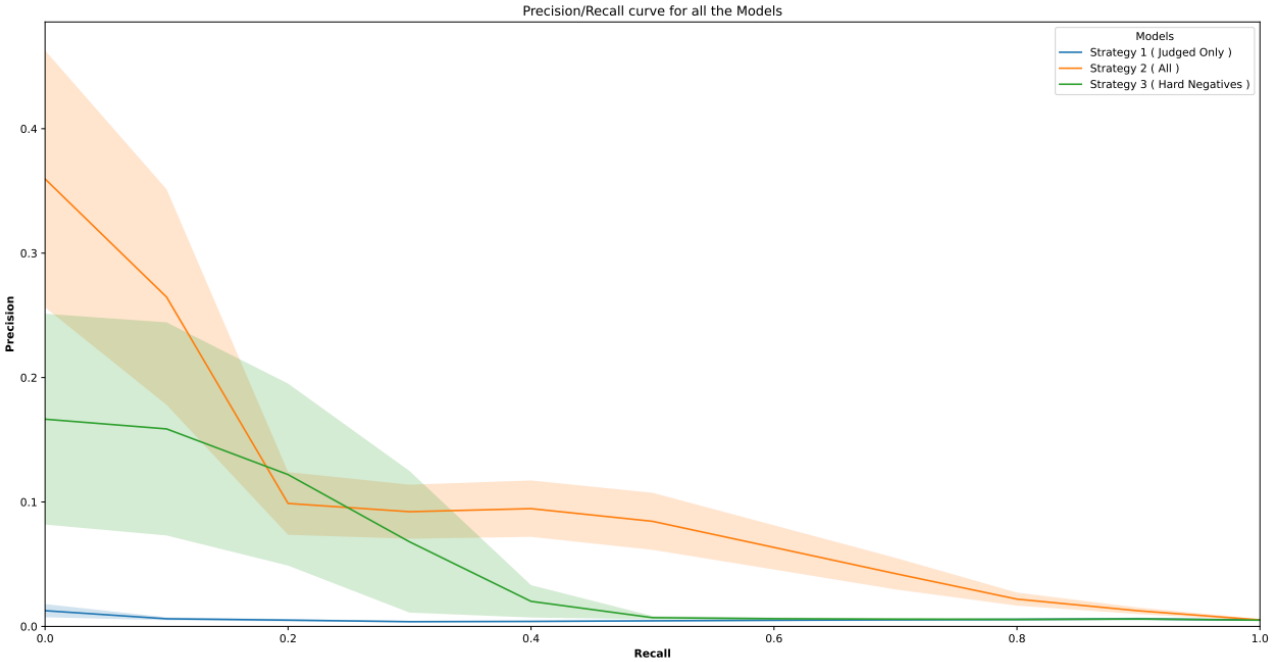

Fig. 4. Precision and recall for each strategy

First it is important to note that queries used to obtain this data visualization are the test queries. From the graphics above we conclude that the LETOR model performed the best, although not great, when using the second training strategy that refers to using all the data for training and consider non-judged documents as non-relevant, which makes sense since this strategy aims to make the model more robust to diverse scenarios, having more data to train the model. When using the Hard Negatives strategy it performed fairly bad, but it performed the worst when using the Judged Only Documents strategy, because it contains less data to train the model, and, for example if we encounter documents that are relevant, but are not judged, this strategy doesn´t categorize them, which decreases the performance. It may not generalize well to unseen documents or situations where relevance judgments are not available. Concluding, since the metrics used vary between 0 and 1, and the obtained results are in the lower half, the LETOR model still has room for improvement.

3.3 Examination of the letor model and discussion of the importance given to each individual feature

Fig. 5. Performance of different strategies applied to hard negative strategy

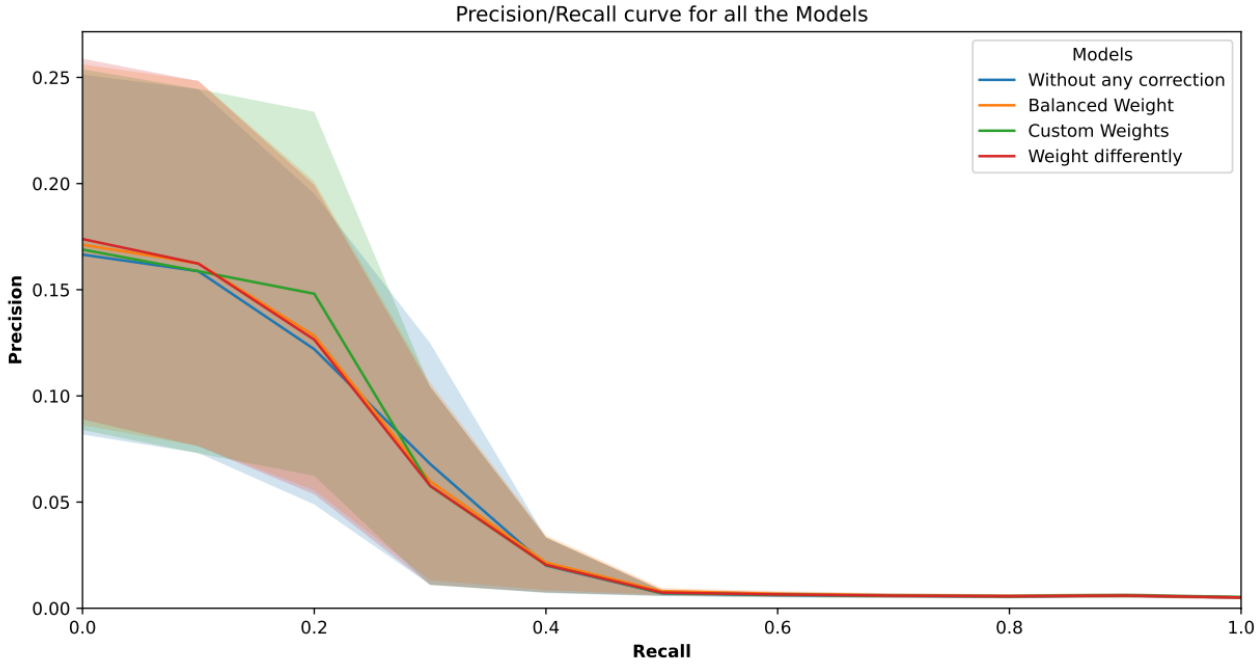

Fig. 6. Precision and recall of different strategies applied to hard negative strategy

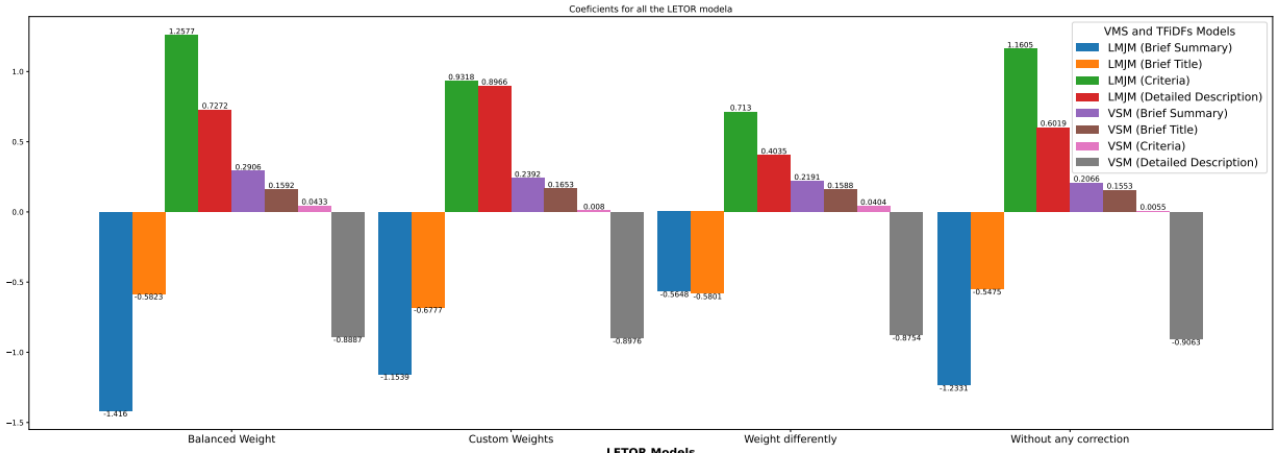

Fig. 7. Coefficients for each different approach to hard negative strategy

By analysing the first bar graph, we can see that the perfomance of the four models to correct data imbalance is somewhat the same, but when using the same weight per class the recall is better, so we concluded that this model is the best one.

Given the weights of the different models from the last graph we can see that the best models were the LMJM models, with the detailed descriptions and criteria, but when analyzing the weights of the VSM models we found very strange that the values were negative for the detailed description and criteria, and we need to run the models more times to see why that happens.

The Custom Weights approach could also be bettered by studying and changing the weights to more appropriate values. We didn't an extensive study of the effects changing the weights of the classes would have.

Its important to note that these models were generated using the second training strategy instead of the judged only documents asked in the project description, because it was the best performing one.

3.4 Analysis on how the training strategies impact the overall results

Regarding the judged only documents strategy, this strategy focuses solely on documents that have been judged and it ensures that the training set consists only of instances where the relevance is known, therefore the model will be trained on a more reliable set of data where relevance judgments are available.

However, it may not generalize well to unseen documents or situations where relevance judgments are not available.

The non-judged documents strategy utilizes all available data for training, treating non-judged documents as non-relevant, this aims to make the model more robust to diverse scenarios. The model may learn to handle situations where relevance judgments are unavailable. However, it might be more prone to noise from incorrectly assuming non-judged documents are non-relevant.

This strategy focuses on challenging instances by using non-relevant/non-judged documents that are ranked high by the initial model, along with all relevant documents, by doing this the strategy aims to improve the model's ability to handle difficult cases and refine its understanding of what makes a document non-relevant. However, it may be computationally expensive, especially if the initial rank is extensive.

4 Qualitative Analysis

4.1 Discussion of the importance of each field according to its content

Regarding the importance of each field we concluded by analysing figure 8 that the most important fields are the description and the criteria of the documents. We believe the description is important due to several reasons:

A detailed description provides more information and context, allowing the models to better understand the content and nuances of the clinical trial. It also provides additional context, background information, and specific details that help the models understand the trial's purpose and methodology. The decip-tion includes a broader range of keywords and specific medical terms that are relevant to a patient case. This specificity allows the models to better match the trial's relevance to the patient's situation. It allows the models to capture semantic relationships between words and phrases. Therefore if the models are trained using detailed descriptions, they learn to associate certain patterns in the language of descriptions with relevance to patient cases.

The criteria can also be relevant because it allows to distinguish if a document is relevant or not.

On the other hand, fields like Brief Tile and Summary are less important due to the opposite reasons, plus, they tend to use abbreviations or acronyms making them ambiguous, increasing the risk of misinterpretation by the models.

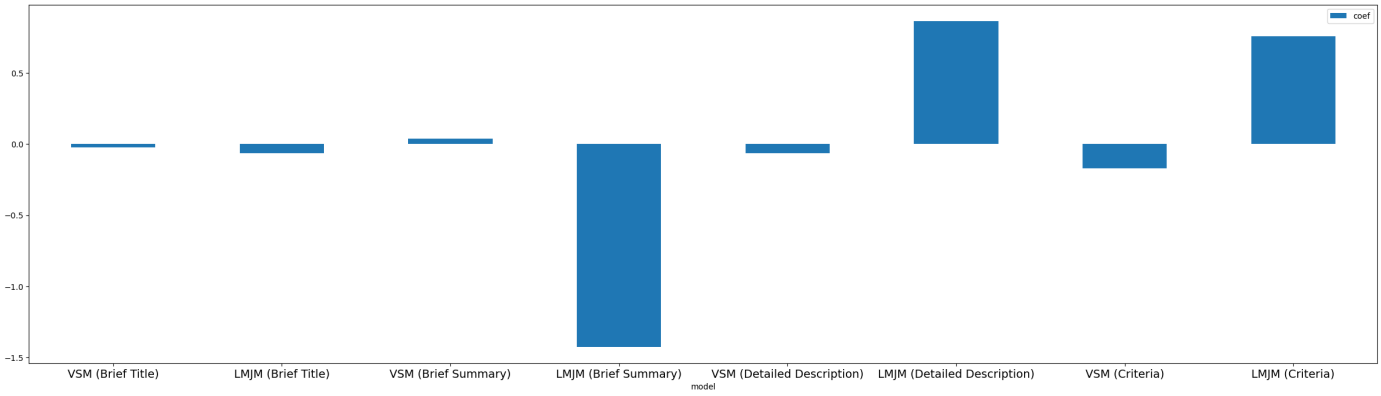

Fig. 8. Coefficients for each pair of corpus and model

4.2 Analysis of failure modes

Regarding the Analysis of failure modes, we concluded that the situation where the system fails are either when we know that a document is relevant but the model is bad and gives a bad output, or vice-versa. Another thing that can contribute to the failure of the system is the fact that we are not normalizing the output of each model. The fact that the LETOR doesn't access the documents is another theory for why the output may not be ideal.

5 Citations

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal articles [1], an LNCS chapter [2], a book [3], proceedings without editors [4], and a homepage [5]. Multiple citations are grouped [1–3], [1, 3–5].

References

1. Author, F.: Article title. Journal **2**(5), 99–110 (2016)
2. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). https://doi.org/10.10007/1234567890
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
4. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
5. LNCS Homepage, http://www.springer.com/lncs. Last accessed 4 Oct 2017