UNIVERSIDAD
POLITÉCNICA
DE MADRID

Information retrieval, extraction and integration

-

Lab ML Ranking Assignment Report

Group 20: Bela Bönte,Dániel Marouf

@github: https://github.com/belaboe97/AdaRank_LOINC

Introduction

The order of results for searches has become very important, as the number of measurements and information is constantly increasing and so the number of answers to a query is also rising. In 2020, it became clear what a pandemic can do and how important it is for the health sector to be able to provide patients with the right care as quickly as possible. To achieve these goals, it can help a lot if researchers can find measurements that are relevant to their topic as fast as possible, and the order of results is one of the key elements due to high search costs.

For a ranking exercise, there are three main options. Pointwise, pairwise and listwise. For this task listwise was used mainly because according to several research studies it is the most effective solution.[1] The solution was based on the research *AdaRank: A Boosting Algorithm for Information Retrieval*[2] and thus the results were ranked using the AdaRank ranking method.

The queries (clucose in blood, bilirubin in plasma, white blood cell count) are optimized using AdaRank. This excercise shows how AdaRank trains a model to produce the best possible results.

[1] https://users.monash.edu.au/~mcarman/papers/ibrahim_TOIS2016_draft.pdf
[2] SIGIR2007_AdaRank.pdf

Dataset

In the field of health, LOINC[3] is a standard language for identifying measurements, observations, and documents related to health. Assume that you are considering an observation as a "question," and that you are considering the observation result value as an "answer".

The results have five columns:

Column Name	Description
loinc_num	The id in loinc system.
long_common_name	The common name of experiment.
component	*The substance or entity being measured or observed.*[4]
system	*The specimen or thing upon which the observation was made.*
property	*The characteristic or attribute of the analyte.*

Original dataset

Each query has seventy possible results. The queries and their corresponding results were imported from the loinc_dataset.xlsx file.

Extended dataset

The extended dataset contains two hundred possible matches.The first seventy matches in the dataset are elements of the original dataset. The remaining part of the extension consists of the hits that were in the loinc search results and were not already in the original dataset. The last part are other random results not related to the query.

Labelling the dataset into relevant and not relevant data

The first task is to tag the query results with a label indicating whether or not the answer was relevant to the query. If it is relevant then the value is 1 if not then the value is 0. Since this is not our field of expertise, the loinc search engine was used to determine the relevant information.

---

[3] https://loinc.org/get-started/what-loinc-is/
[4] https://loinc.org/get-started/loinc-term-basics/

The queries were entered into the search engine and the results were exported. After that, the results were compared with the obtained dataset according to the loinc number and if there was a match, it was considered relevant. Those that were not found in the loinc dataset were considered irrelevant. Since there were a total of six matches for the three queries, the results were checked again individually and the ones that were related to the query were manually set to relevant. In addition, a new id had to be added to the results to link the query to the matches.

For the extended dataset, relevant matches that remain unmatched after a loinc search are assigned a value of 1, while random search matches are assigned a value of 0.

Building parameters for AdaRank

To use and optimize AdaRank, query and text must be converted into machine-readable vectors. These vectors can consist of different key figures. The ranking benchmark dataset Letor 4 is used as orientation[5]. In this dataset, term frequencies and IDF scores are used as parameters. Therefore, TF were also formed for the search in Loinc. In addition, similarity scores between query and texts were formed using the Spacy model en_core_web_lg and the NLP modul. Just constructing these metrics for the match between query and searched text respective Loinc entry is crucial. Since this is a concept of proof, only 4 metrics were developed. The benchmark set Letor4 has over 40 such metrics.
The metrics used are as follows:

- Term Frequency Query - (Loinc) Long Common Name
- Term Frequency Query - (Loinc) Component
- Similarity Score with Spacy NLP Model Query - (Loinc) Long Common Name
- Similarity Score with Spacy NLP Model Query .- (Loinc) Component

AdaRank - A Listwise approach

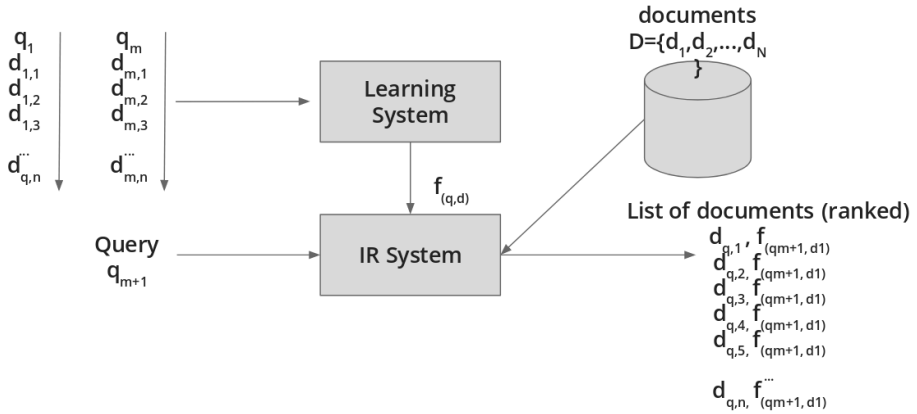
Listwise techniques look at the complete list of documents and attempt to determine the best way to arrange them in the most efficient manner. When compared to pointwise or pairwise techniques, listwise approaches can be considered more sophisticated.[6] In contrast to pointwise approaches, where only the match of the query to a document is calculated and then ranked, there are pairwise approaches. These distinguish binary whether the ranking to the following document is correct.

With Listwise Approaches, the entire ranking is considered. This is important because the optimization of the models differs. AdaRank assigns a weight to each parameter. Based on the list and the labels, AdaRank can now optimize the recommendations. To learn, AdaRank creates weak learners. A linear combination of "weak rankers" is used as the basis for AdaRank. As part of the learning process, it repeats the

[5] https://arxiv.org/ftp/arxiv/papers/1306/1306.2597.pdf
[6] https://medium.com/@nikhilbd/pointwise-vs-pairwise-vs-listwise-learning-to-rank-80a8fe8fadfd

steps of re-weighting the training sample, producing a weak ranker, and computing a weight for the weak ranker again and again.[7]

Figure 1 : Listwise Approach of a ranking system



Scoring for AdaRank

AdaRank can use different scoring metrics and optimize them. Normalized Discounted Cumulative Gain (nDCG) is used in this case study. “Normalized Discounted Cumulative Gain (NDCG) is popular method for measuring the quality of a set of search results. It asserts the following: Very relevant results are more useful than somewhat relevant results which are more useful than irrelevant results (cumulative gain)”[8]
The first step is to see if the result list is relevant.However, this does not include which results are displayed first. Therefore, a metric is introduced that recognizes when relevant results are displayed over irrelevant ones, called DCG. Since it is harder to find relevant results for some queries than for others, the results can be normalized afterwards. Based on this score, the weights of the parameters in this AdaRank model are reweighted.

Results

The effectiveness of the architect is shown by the order in which the relevant results appear after sorting. It works well if it prioritises the results that are relevant to the query and ranks them accordingly. In total, six rankings of the results were produced. Two for each query, one for the original training set and one for the extended set.

Since this model is an existing implementation, an error occurred in the case study. As soon as several queries (marked by different qids) were to be processed, an error occurred. However, the data set Letor 4 was also tested with which different querie results could be

[7] SIGIR2007_AdaRank.pdf
[8] https://blog.thedigitalgroup.com/measuring-search-relevance-using-ndcg#:~:text=Normalized%20Discounted%20Cumulative%20Gain%20(NDCG,than%20irrelevant%20results%20(cumulative%20gain)

optimized at the same time. The results are therefore an nDCG respectively DCG optimization of single queries.

The results consist of the following three attributes:

- *Loinc Number*
- *Rank*
- *Matching score*

First query: Glucose in blood

Table 1: Results for Glucose in blood

Original	Extended
18928-2 1 0.09133297048224612 18906-8 2 0.07025613114018932 18865-6 3 0.04566648524112306 6768-6 4 0.04566648524112306	40650-4 1 0.46487346296660087 58409-4 2 0.46484876708422546 33837-6 3 0.45666132496965667 8357-6 4 0.44277312457684936

Since here was the least relevant hit on the original dataset, the ranking was sorted with very inaccurate results. The dataset extended by the loinc search gave much better results because 70 relevant results were added to the dataset.

Second query: Bilirubin in plasma

Table 2: Bilirubin in plasma

Original	Extended
3082-5 1 0.0730663763857969 2039-6 2 0.048069984464340056 1003-3 3 0.03382702610453559 882-1 4 0.0001261045359406324	33898-8 1 0.10245924264225102 35191-6 2 0.08762031784578708 18264-2 3 0.08469964058426084 18878-9 4 0.04886517726015048

In the second case, there are far fewer relevant matches in the extended dataset, so the differences between the results are not so significant.

Thrid query: White blood cells count

For the third query, the results for the original data set are 0 and for the extended dataset the rating jumps to infinity. This seems to be an error, which could not be found in the code or in the Excel sheet. Nevertheless, when comparing the results, it was noticed that results labelled with Relevant (1) were ranked before non-relevant results (0).

Table 3: White blood cells count

Original	Extended
3 Q0 2093-3 1 0.0	14383-4 1 inf

3 Q0 18878-9 2 0.0 3 Q0 2028-9 3 0.0 3 Q0 1994-3 4 0.0	75352-5 2 inf 70039-3 3 inf 76298-9 4 inf

As already mentioned, the circumstances described above make it difficult to draw conclusions, but empirical experience has shown that the queries of the validation set seem to be correctly ranked.

Conclusion

As it can be seen, when sorting the extended training set, most of the relevant hits are placed at the top of the order while the smaller original learning set does not yet work effectively enough.

Furthermore, because there were very few relevant matches in the original dataset, the ranking algorithm did not have enough examples to learn from. Eliminating this in the extended dataset where more relevant hits were added to the original dataset gave better results.