Lab ML Ranking Assignment

| Group Member |
| --- |
| Julio Nevado Delgado |
| Wenqi Jiang |
| David Cabornero Pascual |

# 

# Introduction

Many information retrieval tasks, such as document retrieval, online advertising, collaborative filtering and sentiment analysis, include ranking as a major component. The application of machine learning, like supervised, semi-supervised, or reinforcement learning, in such ranking models of Information Retrieval Systems is known as Learning to Rank or Machine Learned Ranking (MLR). The purpose of building these ranking models is to rate fresh and unseen documents in the same way that the training data was ranked.

Tie-Yan Liu of Microsoft Research Asia has analyzed existing algorithms for learning to rank problems in his book *Learning to Rank for Information Retrieval*.[1] He categorized them into three groups by their input spaces, output spaces, hypothesis spaces (the core function of the model) and loss functions: the pointwise, pairwise, and listwise approach. In our assignment, we used the most efficient and common approach — pointwise, because it is fast to train and predict, and there are many different machine learning models that can be implemented by using it, like Logistic Regression, Linear Regression, Naive Bayes. So we built a logistic regression model to predict the queries’ relevant level.

## Dataset

For the training data, we prepared more than 200 documents as well as queries, which are “glucose in blood”, “bilirubin in plasma” and “White blood cells count”. Additionally we extended the original data to a total of 360 rows of data for training modeling.

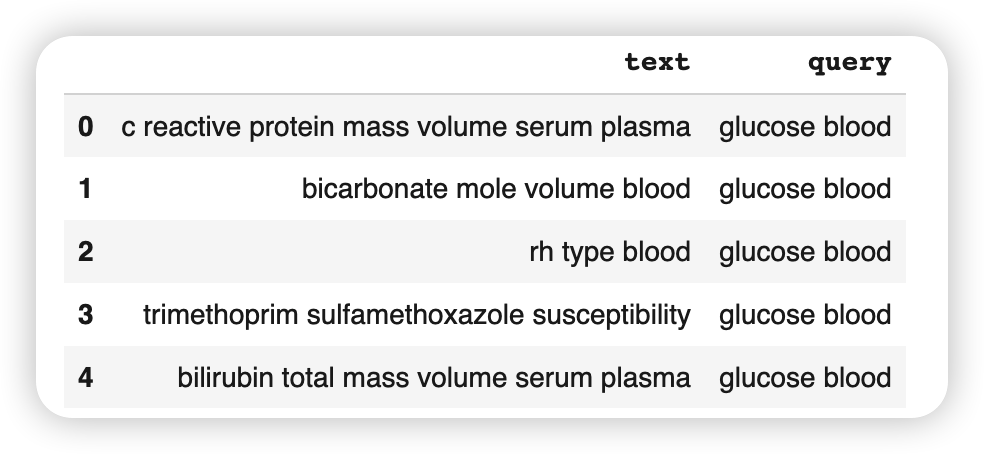
# Modeling

## Preprocessing

### Text Cleaning

After importing the raw data, the first task is to clean the data, here we use the package Natural Language Toolkit(NLTK) to do this job. NLTK is a leading platform for building Python programs to work with human language data. It provides a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries[2].

After removing punctuation and stop words, lemmatizing, tokenizing and other preprocessing steps by using NLTK, we came up with the following dataset.



### Dataset Generation

Dataset is composed of document - query - term triplets where the term is contained both, in the query and in the document. For each triplet, several attributes that build up the dataset are computed. These terms are:

* Query absolute frequency (QAF): Number of times the term appears in the query.
* Query relative frequency (QRF): QAF divided by the number of terms in the query.
* Document absolute frequency (DAF): Number of times the term appears in the document.
* Document relative frequency (DRF): DAF divided by the number of terms in the document.
* Inverse document frequency (IDF): Number of times a term appears in the whole collection divided by the number of documents.
* Relative frequency in all documents (RFAD): Number of times a term appears in the whole collection divided by the total number of terms in the collection.

Following the previous, for the following triplet:

* Document: bicarbonate moles volume blood
* Query: glucose blood
* Term: blood

The computed attributes would be:

* QAF:
* QRF:
* DAF:
* DRF:
* IDF:
* RFAD:

Additionally, specifying relevance for each query - document pair is needed. Since for the provided loinc dataset documents are the same for the three queries but in a different order, we assume that that ordering contains the relevance. Therefore we establish a ranking based on the order.

However, since the implemented model is logistic regression which requires binary relevance judgements, a relevance of 1 (relevant) is set for the 20 most relevant documents for each query and a relevance of 0 is set otherwise.

## Model Implementation

The implemented model is logistic regression with the previously mentioned 6 attributes. The model is as follows:

, where *R* is the probability that a sample is relevant and *Odds* is the odds ratio of the logistic regression estimation. The previous six terms are logged in order to smooth them. It can be seen that the previous formula computes the logarithm of the relevance for a query - document - term triplet. However, we would be interested in the relevance given a query - document term.

The desired result is obtained by summing out over terms contained in both, query and document:

Only thing left to do is depicting the obtention of prior, *Odds(R)*. This number is the odds ratio of the estimation that our document is relevant to our query, but taking only into account the query itself. With only this information, the probability of being relevant is the number of relevant documents for this query divided by the total number of documents.

## Dataset Extension

In order to extend the dataset two additional queries are performed over LOINC:

* Glucose in urine
* Urate in urine

The obtained results are exported to CSV format. Again, relevance is inferred from the ordering in the results, being the 20 first results relevant on the rest considered non relevant.

Conversion to the six previously mentioned attributes is performed in the same way as for the proposed dataset.

# Conclusions

In this study, we created a document search strategy based on a logistic regression model using a pointwise approach. This method of logistic inference calculates the likelihood of documents being relevant to a query that represents the user's information requirement. In the real world test document – “Urate [Mass/volume] in 2 hour Urine” and query – “urate in urine", we got 0.6993 as the relevant result which was quite good compared to traditional models. Besides building the logistical model, we extended our training dataset performed over LOINC. We got a greater grasp of how the process works in the real world, as well as the necessity of ranking documents, and we increased our capacity to find information in practice.

# References

[1] Tie-Yan Liu (2009), "Learning to Rank for Information Retrieval", Foundations and Trends in Information Retrieval, 3 (3): 225–331, doi:10.1561/1500000016, ISBN 978-1-60198-244-5. Slides from Tie-Yan Liu's talk at WWW 2009 conference are available online Archived 2017-08-08 at the Wayback Machine

[2] Nltk.org. 2022. NLTK :: Natural Language Toolkit. [online] Available at: <https://www.nltk.org/> [Accessed 13 April 2022].