Data Mining Task 2

Frank William Hammond Espinosa
Maria Àngels Llobera Gonzàlez

22nd April 2025

1 Introduction

In this work, we attempt to assist a hypothetical biologist by summarizing and semantically clustering 1034 different biological functions that affect 20 bacteria. To do so, we use a natural language processing pipeline that leverages a pretrained Transformer from the HuggingFace Hub named "Bio-BERT" [**Lee˙2019**].

The structure of the document is the following: first, we will explain the methodology used, highlighting both the Data preprocessing and model implementation; then, some results will be presented and discussed. Lastly, we will state the conclusions of this work.

2 Methodology

The code used in this assignment can be found in https://github.com/Frankwii/data_mining_2.

2.1 Data and preprocessing

As input files, we have texts with the functions that affect 20 bacteria. These have been preprocessed and converted to JSON files in order to be able to work with them more easily. Results can be found under the 'resources' folder in the companion repository.

2.2 Model and code

We have improved on the results of the partial delivery of 8th April. First, the code has been restructured using object-oriented programming.

2.2.1 Clustering and ranking

We have continued our efforts about ranking the obtained function clusters, extending the previous approach in a nontrivial way. Inspiration was drawn from the classical technique used in document and image retrieval which leverages the so-called Term Frequency-Inverse Document Frequency (TF-IDF) scoring mechanism. It is perhaps best understood in the context of document retrieval:

Let $D_1, \dots, D_N$ be a sequence of multisets which we will call *documents* all in the same language L, such that the set of words in L is given by $w_1, \dots, w_M$. The **term frequency** of the word w_i in document

D_j is given by

$$f_{ij} = \frac{|\{w \in D_j \colon w = w_i\}|}{|D_j|}$$

(recall that the D_j are multisets; this definition would be trivial otherwise). Additionally, the **inverse document frequency** of word w_i is given by:

$$\zeta_i = ln\left(\frac{N}{|j \colon w_i \in D_j|}\right).$$

This definition is ill-posed for small documents or large vocabulary, and hence replaced by the numerically-friendlier version $ln\left(\frac{N+1}{|j: w_i \in D_j|+1}\right)$.

Finally, the **TF-IDF** of word w_i in document D_j is given by

$$\rho(i,j) = f_{ij} \cdot \zeta_i$$

An adaptation of this approach was carried out in our work: we thought of bacteria as words and of clusters of functions as documents: call B_i the set of bacteria affected by function ϕ_i. If cluster C_j is formed by functions $\phi_{i_1}, \ldots, \phi_{i_{n_j}}$, we define the document D_j as

$$D_j = \bigcup_{k=1}^{n_j} B_{i_k}$$

and word ϕ_j simply as the function ϕ_j. Now, a simplification is made to the original algorithm: instead of computing the relative frequencies of words in specific documents, we count the relative frequencies of words overall. That is, in our case f_{ij} is substituted by the the number of bacteria that are affected by function ϕ_j divided by the total number of bacteria, which in our case is 20. Notice that this does not depend on i.

Finally, we can regard $\rho(i,j)$ as a matrix $P = (\rho(i,j))_{i,j}$ which can be exploited in different manners in order to sort our function clusters. When given a document D_j, we consider only those rows of P which refer to bacteria in D_j and then aggregate them in one of a few different ways to obtain a score, which is then used to sort clusters decreasingly.

We implemented all 9 possibilities of row/column aggregations combining the minimum, maximum and average. For us, it made the most sense to maximize by columns and then minimize by rows, since that would mean that most functions of the cluster are bacterium-wise rare.

2.2.2 Semantic search

We have implemented a **semantic search engine** that allow ranking either biological function or bacteria based on the semantic similarity between a query text and the function database.

For biological functions, this has been achieved by using the previously calculated embeddings, embedding the query text itself and then ranking biological functions in the database by different similarity functions such as cosine, dot product and Pearson correlation.

In addition, we can extend this approach to rank not only biological functions but also bacteria, which in this context can be seen as clusters of functions. Thus, traditional clustering similarity algorithms can be exploited: we can run different search criteria, considering similar bacteria with the affected biological functions similar on average, or looking for outliers with a maximum or minimum.

3 Results and discussion

Here are some results after applying *min-max* aggregation to sort the clusters:

```
========CLUSTER 1========
Amino sugar and nucleotide sugar metabolism

Affected bacteria: Pseudomonas putida, Rhodopirellula baltica,
Clostridium botulinum, Pseudomonas aeruginosa, Escherichia coli

========CLUSTER 2========
Mixed, incl. MIP18 family-like, and Armadillo

Affected bacteria: Trichormus variabilis

========CLUSTER 3========
Oxoacid metabolic process

Affected bacteria: Geobacter sulfurreducens

========CLUSTER 4========
Mixed, incl. polysaccharide metabolic process, and extracellular region

Affected bacteria: Vibrio cholerae

========CLUSTER 5========
Catalytic complex
Catalytic activity

Affected bacteria: Trichormus variabilis, Neisseria meningitidis,
Pseudomonas putida, Rhodopirellula baltica, Clostridium botulinum,
Streptococcus pneumoniae, Listeria monocytogenes,
Staphylococcus aureus, Yersinia pestis, Chloroflexus aurantiacus,
Salmonella enterica, Vibrio cholerae
```

Table 1 shows different results for a simple query, which is part of the name of some biological functions in our dataset. We see promising and curious results, since by applying different aggregation methods, we obtain different results.

4 Conclusions

This project demonstrates the practical application of modern NLP techniques for complex tasks involving biological data. By embedding and clustering textual information of affected functions across 20 bacterial species, we created meaningful clusters that might help a biologist quickly understand the biological impact of antibiotic treatments.

We faced difficulties assessing the performance of some of the methods proposed, since that would require close collaboration with experts on the field. However, we are confident that with some fine-tuning of the different algorithms proposed for clustering and retrieval, and an optimal implementation in scale, these methods could be of use to biologists dealing with large datasets of bacteria and functions.

Bacteria	**Average**	**Maximum**	**Minimum**
Escherichia coli	**0.8641**	0.9758	0.9130
Yersinia pestis	0.8578	**0.9804**	**0.9500**
Shigella flexneri	0.8550	0.9604	0.8990
Bacillus subtilis	0.8519	0.9758	0.9045
Streptococcus pneumoniae	0.8507	0.9322	0.8931
Pseudomonas putida	0.8507	0.9804	0.9075
Aquifex aeolicus	0.8505	0.9758	0.9500
Neisseria meningitidis	0.8491	0.9376	0.9500
Geobacter sulfurreducens	0.8491	0.9758	0.9500
Sinorhizobium meliloti	0.8491	0.9322	0.9500
Salmonella enterica	0.8458	0.9426	0.9014
Pseudomonas aeruginosa	0.8446	0.9604	0.9500
Clostridium botulinum	0.8446	0.9804	0.9354
Trichormus variabilis	0.8413	0.9433	0.9373
Staphylococcus aureus	0.8359	0.9405	0.9314
Vibrio cholerae	0.8258	0.9804	0.9335
Chloroflexus aurantiacus	0.8204	0.9332	0.9373
Helicobacter pylori	0.8140	0.9027	0.9001
Rhodopirellula baltica	0.8084	0.9386	0.9500
Listeria monocytogenes	0.7837	0.9322	0.9130

Table 1: Results after applying semantic search for the query "*Flagellar*" using cosine similarity function and different aggregation methods. In bold, the highest score for each aggregation method.