Plants

Information Processing and Retrieval

Bárbara Rodrigues
Faculdade de Engenharia
da Universidade do Porto
up202007163@edu.fe.up.pt

Rúben Monteiro
Faculdade de Engenharia
da Universidade do Porto
up202006478@edu.fe.up.pt

Sofia Costa
Faculdade de Engenharia
da Universidade do Porto
up202300565@edu.fe.up.pt

Tiago Ribeiro
Faculdade de Engenharia
da Universidade do Porto
up202007589@edu.fe.up.pt

ABSTRACT

In a world overflowing with information there is the demand to find relevant data that aligns with a user's information needs.

This report describes methods for retrieving and processing data about various plant species, scrapped from Wikipedia. After thorough data cleaning and preparation, the resulting dataset underwent exploration, revealing compelling and noteworthy insights. Furthermore, the data was indexed for use by a data retrieval tool, Solr, and the results obtained were evaluated and compared on their precision between different systems. The work done made it possible to enhance the search for various plant species and make sure relevant results for a user's search task were successfully retrieved.

The search system developed is intended to assist those interested in finding specific plants that match their information needs.

KEYWORDS: data processing, data retrieval, scraping, data evaluation, data characterization, search system, plants, semantic search

1 INTRODUCTION

Within the scope of Information Processing and Retrieval, this report describes the work done for a three-milestone project during the fall semester. By the end of the third milestone, a functioning search system for plants has become available.

Following an extensive search for an interesting dataset rich in textual data, it was decided as a group that the topic for the project would be "plants". Subsequently, the data related to this topic was first collected, prepared, explored, and processed by applying the concepts acquired throughout the semester.

During the information retrieval phase, the data treated in the previous section was moved to Solr collections, a system where it would be subjected to queries, and its results evaluated afterward. The results were evaluated based on the precision of retrieving relevant plants for the query formulated. To possibly improve the search system, the introduction of semantic search with the use of embeddings in the system, was tested and evaluated on its retrieving performance. This was believed to improve the results obtained since it can understand the context of a user's search task rather than relying only on keyword matching.

2 DATA PREPARATION

The first milestone of the project, focused on the preparation and characterization of the data. This phase is highly dependent on the chosen topic's dataset which required extraction actions such as scraping.

The expected outcome from this initial phase is a well-documented and reproducible pipeline for data processing to simplify and enable subsequent milestones.

Additionally, in this phase, after the data was prepared and transformed into a well-organized dataset, a few exploration tasks were performed to draw interesting insights about the data at hand.

2.1 Data Selection

After having trouble deciding on a diverse range of prepared datasets, such as music lyrics or wine reviews, a consensus was reached collectively that the topic of the project would be focused on plants. This is a very text-rich and well-documented theme that is widely and easily available to anyone researching it.

With this topic in mind, it was not hard to find an authentic source, as Wikipedia, the widely known encyclopedia, already had many wide lists full of information, ready to gather and use. The specific list is "List of plants by common name" [2]. Wikipedia content is typically available under a Creative Commons license, allowing for the use and redistribution of the data.

Each list entry provides a table of simple scientific information regarding the plant, along with multiple and diverse paragraphs written about the plant's characteristics and details. The list bears around 400 unique values, each one containing one or more paragraphs with information about the plant.

Finally, to collect all this data, the gathering process chosen was web scraping. By making use of Python [3] scripts and the "Beautiful Soup" library [4], it was possible to collect all the information relative to each plant of the list that had a link attached to it.

The information extracted was then stored in a CSV file, ready for the next part of the pipeline.

2.3 Data Processing

Upon completing the data collection phase and analyzing the final data of the dataset, a series of processing tasks were executed to improve the quality of the data. All the steps taken to reach a clean and complete dataset are illustrated in **Appendix A.1**, which represents the structured and reproducible pipeline of the project.

Firstly, the incongruence in column names was noticeable. For example, some column labels end in ":" such as the "Kingdom" column which appears as "Kingdom:", so all these values were rewritten to appear with the same representation.

Another challenge faced was the inconsistent organization of data across different articles. Specifically, similar types of information were often presented in varying formats and headings. For instance, columns "Habitat", "Habitat and range" and "Habitat and distribution": those columns were aggregated and combined to make only one, in this specific case, the "Habitat" column. This was the main step to ensure the homogeneity of the dataset. These variations were merged and reconciled, harmonizing the data by aligning these sections under a common label. Initially, the dataset had 178 columns but after merging and transforming, it was reduced to only 63 columns.

Following data aggregation involved removing irrelevant columns for future exploration. These columns either had a large number of "NaN" values or offered a very limited amount of data, leading to the removal of 28 unnecessary columns.

Upon closer examination of certain column contents, namely the "Introduction" and "Description" columns, details about the geographical origin of the plants were uncovered. Given this discovery, it was decided to extract this information and organize it into a fresh column named "Origin Country".

Preceding saving the transformed dataset, the "Name" column was modified in only three cases. In contrast to other samples where the "Name" column had just the plant name, these specific instances had lengthy text which is believed to be due to a scrapping error. These cases were manually adjusted to include only the original name by reading the "Name" field.

The data processing phase was essential for creating a cohesive and structured dataset, facilitating subsequent analysis, and ensuring that our data remained consistent and easy to work with. Therefore, the cleanup, processing, and transformation of the original dataset was completed, resulting in a final set of 36 columns and 403 rows containing information that is believed to be crucial for the project's ongoing progress.

The conceptual model for this dataset can be observed in the following figure (**refer to Figure 1**):

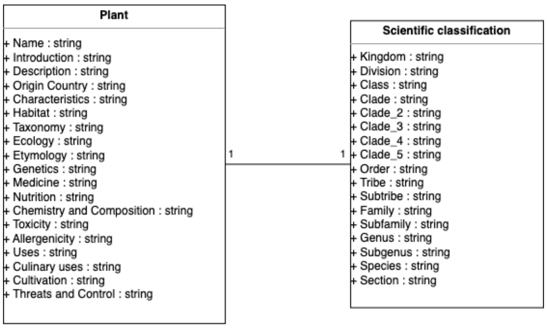


Figure 1 - Conceptual Model

2.4 Data Characterization

Following the data cleaning and refinement process, an exploration analysis of the dataset was conducted, to gain a deeper understanding of the data extracted. Using Jupyter Notebooks [5], it was possible to create plots such as bar plots, pie charts, and word clouds for the entire dataset. The number of rows in the dataset is equivalent to the number of plant species scrapped from the Wikipedia list [2].

Exploring the Scientific Classification of plants seemed like an interesting task to retrieve some insights about the dataset. This classification system includes groups such as Kingdom, Clade, Order, Family, Genus, Species, and other subgroups.

Table 1 - Count of unique values for the Scientific Classification column

Group	**Count**
Kingdom	1 (*Plantae*)
Clade	1 (*Tracheophytes*)
Order	43
Family	87
Genus	203
Species	299

Focusing on the "Order" and "Family" groups **(refer to Table 1)**, since they had smaller values to work with, two bar plots were generated to illustrate the distribution of plant species for each value within each of these classifications.

By observing **Figure 2**, it can be concluded that the "Order" with most plant species is called "*Asterales*", which according to Wikipedia, brings together "dicotyledonous plants, like marigolds, daisies, and sunflowers" [6]. Additionally, according to **Appendix A.2**, the "Family" with more plant species is "*Asteraceae*" which "consists of over 32,000 known species of flowering plants in over 1,900 genera within the order *Asterales*" [7]. These results suggest that flowers are the dominating plant of the dataset.

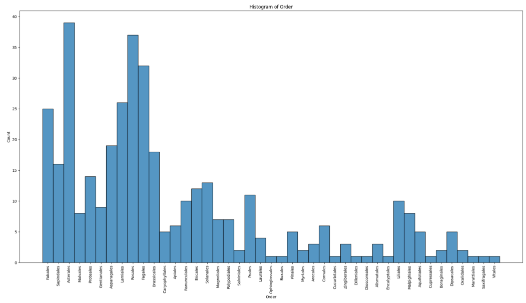


Figure 2 - Barplot for the 'Order' column

Since every column of the dataset is rich in textual data, the data exploration was limited to text analysis. Therefore, word clouds were generated, for any relevant column, as well as finding out the frequency of specific words.

By generating a word cloud for the column related to the plant's name, it can be concluded which species dominates the list.

By analyzing **Figure 4**, four notable *Genus* within the "Name" column are identified, specifically *Quercus*, *Lambertia*, *Allium*, and *Acer*. These designations correspond to the first words in species names, representing the respective genera. *Quercus*, also known as oak trees, is observed 17 times in the dataset. *Lambertia*, recognized as "wild honeysuckle," appears 10 times. *Allium*, the Latin term for garlic, is recorded seven times. Lastly, *Acer*, the genus for maples, is documented six times in the respective column.

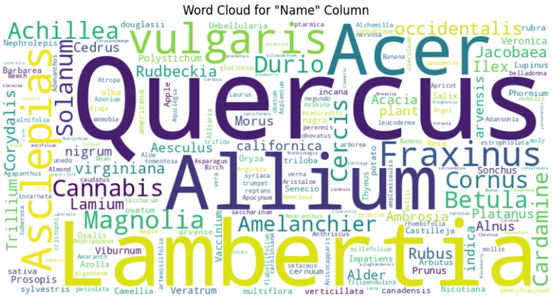


Figure 3 - Word Cloud for the "Name" column

More word clouds were generated for each column but only some presented relevant results. By generating a word cloud for the column related to each plant's origin country (refer to Appendix A.3), it was possible to identify the continent or country where most plants originated from. North America and Australia were the most mentioned native locations.

As for the medicine column (**refer to Appendix A.4**) the words that stand out the most are "antioxidant" and "disease" which can imply the connection of plants with potential therapeutic benefits and disease prevention.

Since plants can have multiple different characteristics, an interesting task for data analysis was to extract the number of mentions of specific colors. The generation of a pie chart **(refer to Figure 7)** with the count of each color mentioned led to the identification of the most predominant colors in the whole dataset, which are: red, green, and white.

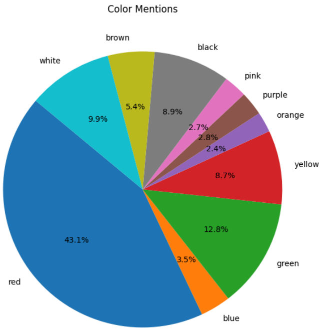


Figure 4 - Pie Chart of color mentions count

2.5 Prospective Seach Tasks

In this section, a few prospective search tasks are outlined for a comprehensive plant information search system. Each task aims to address specific information needs the user might have, to make their search experience easier and more efficient.

- **Which plants can survive in cold environments?**
- **Which plants have purple flowers?**
- **Which plants are trees and have edible fruits?**
- **Which plants are associated with Christmas?**

3 INFORMATION RETRIEVAL

The information retrieval phase of this project focuses on obtaining relevant information from the dataset centered around plants. The aim is to address users' specific information needs, such as the search tasks specified before, effectively.

For this purpose, Solr [8] was the recommended information retrieval tool to be used. Its selection was based on its robust search capabilities, scalability to handle large datasets and efficient search functionalities.

The process taken for this phase is described in the diagram presented in **Annex A.5**.

3.1 Collection and Indexing

At the end of the preparation phase, the final dataset was in a CSV format however, to populate the collection created in Solr, this data was converted into a JSON file. This conversion facilitates efficient indexing and retrieval of data when using the tool since it aligns with the document structure in Solr, providing a faster and smoother retrieval. Within the created collection, each document represents a plant, characterized by a set of specific fields.

The indexing process started by deciding which fields were relevant to be indexed and which were not. If indexing is true, the field's value can be used in queries to retrieve matching documents. All fields are rich in text, some only have single-word elements, such as the fields related to the plant's scientific classification, and others have very extensive text. Given the text richness in each field, it was concluded that all fields were relevant to be indexed, except the 'Kingdom', given that it only has one unique value.

The indexed fields were added to the schema file and are described by their name, type, indexed, and stored. An example of the schema field types can be described in the following table:

Table 2 - Example of three schema fields with the three different field types

Field	**Type**	**Indexed**	**Stored**
Name	string	Yes	Yes
Introduction	text	Yes	Yes
…	…	…	…
OriginCountry	commaSep	Yes	Yes

The fields rich in extensive text and sentences were given the custom field type of "**text**" and the fields described by one or two words were given the existing Solr field type "**string**". Additionally, for the fields that have comma-separated designations, like in the 'Origin_Country' column, a new field type called "**commaSep**" was created.

The "text" custom field type uses the "**StandardTokenizerFactory**" token [9]. To the index analyzer of this field type were added the following filters [10]:

- **ASCIIFoldingFilterFactory**: Performs ASCII folding. Converts accented characters to their ASCII equivalents.
- **LowerCaseFilterFactory**: Converts all letters in the text to lowercase.
- **SynonymGraphFilterFactory:** Improves search recall by ensuring that searches for one term also match documents containing its synonyms.
- **PorterStemFilterFactory**: Performs stemming to reduce words to their root form. It is only appropriate for English language text.
- **RemoveDuplicatesTokenFilterFactory**: Removes duplicate tokens in the stream, i.e. the ones that have the same text and position values.

A slightly different approach in the "commaSep" custom field type is that it uses the "**PatternTokenizerFactory**" token, which, in this case, breaks the input text stream into tokens by the comma separation. The filters **ASCIIFoldingFilterFactory** and **LowerCaseFilterFactory,** previously described, were also added to the index analyzer of this field type.

3.2 Retrieval

After the indexing process is complete, the next step was to decide which queries to perform to retrieve the information needed for each search task.

All four information needs were described with the given context, and for each, one simple and one boosted query were defined using the eDisMax query parser.

Using eDisMax [11] for each search task, the following parameters were used:

- **q** (Query) is the main query parameter.
- **q.op** (Query Operator) specifies the default operator for query expressions.
- **qf** (Query Fields) lists the fields to search, with a boost factor. Fields with higher boost factors contribute more to the overall score.

To improve the search results and retrieve the most relevant information, a few search ideas were explored, such as term boosts, field boosts, and proximity searches.

The use of term and field boosts allowed the assignment of a higher importance to specific keywords or phrases which would possibly improve the ranking of the search results. Additionally, the proximity search allowed the narrowing down of search outcomes by considering the closeness of terms by a given number of words, ensuring they were related.

Three systems were defined to evaluate the document retrieval performance:

- System 1: **Schemaless**. Queries were executed without the use of a schema.
- System 2: **Simple system**. Applied the schema described previously and executed simple queries without applying any boost.

- System 3: **Enhanced system**. Applied the schema described previously and executed the same query with different kinds of boosts, mainly field boosts and proximity boosts.

With this, it was possible to verify if the use of a schema and enhancing the system would improve the precision of the search results.

3.2.1. Search Task 1 - Which plants can survive in cold environments?

Information Need: A museum is considering opening a botanical garden in its outdoor area, but located in a high latitude area, they must be aware of the climate in the winter. Before consulting an expert, they decide to search for plants that could handle it to have a general idea.

The information about this characteristic of a species can be presented in many ways, not only using the word "cold", but with references to snow or ice, or describing its resistance to freezing, so the use of synonyms was crucial.

Simple Query:

- q: "low temperature" "benefic cold" "tolerant cold" "remains cold"
- q.op: OR
- qf: Introduction Description Characteristics Ecology

Boosted Query:

- q: "low temperature"~3 "benefic cold"~3 "tolerant cold"~3 "remains cold"~3
- q.op: OR
- qf: Introduction^2 Description^3 Characteristics Ecology^0.7

Relevance Judgment: For this query, it was important to only retrieve plants that mention they can survive in cold environments or in low temperatures.

3.2.2. Search Task 2 - Which plants have purple flowers?

Information Need: Let's imagine a situation where a user is going for a hike, during this activity he notices a purple flower that he has never seen before. The curiosity to identify it is exactly what we have in mind for this information need.

When searching in the dataset, it was discovered that 'flowers' can also be referred to as 'blooms', which is why that term was defined as a synonym for 'flower'. The same was applied to 'violet' and 'lilac' which have the same meaning as 'purple'.

The word purple was given the most importance since it is the main goal of this query. The word 'flower' will appear very frequently in the dataset and not necessarily be useful, but purple will be rarer in a context outside of 'flower'. It is also prioritized when 'purple' appears neighboring the words 'flower' or 'bloom'.

Simple query:

- q: purple flowers species
- q.op: AND
- qf: Introduction Description Characteristics

Boosted query:

- q: "purple flowers"~5 species purple^5
- q.op: AND
- qf: Introduction Description^3 Characteristics

Relevance Judgment: For this query, it was important to only retrieve plants that could be either flowers or trees with flowers that are purple.

3.2.3. Search Task 3 - Which plants are trees and have edible fruits?

Information Need: For his garden, a user is planning on planting a tree, besides decorative purposes and possible shade, the user is considering one that may provide him with edible fruits. Wish that inspired this information need.

Since the key in this search is the presence of fruit, this word and "edible" are boosted and combined with the word tree.

Simple query:

- q: fruit trees edible
- q.op: AND
- qf: Description Introduction Characteristics

Boosted query:

- q: fruit^5 trees edible^2
- q.op: AND
- qf**:** Description^2 Introduction Characteristics

Relevance Judgment: For this query, it was important to only retrieve plants that are fruit trees and whose fruits are edible.

3.2.4. Search Task 4 - Which plants can be used as Christmas decorations?

Information Need: The owner of a garden center wants to take advantage of the holiday season to sell more. He wants to gather information on what he can sell as Christmas decorations.

The word "Christmas" is the most important as the season is the central item of the information need.

Situations where Christmas appears close to words like "celebration", "decoration" or "season" are boosted to show priority concerning lone terms that are related to the subject, i.e. "tradition" and "ornamental".

Simple query:

- q: Christmas celebration decoration season tradition ornamental
- q.op: OR
- qf: Introduction, Description, Etymology

Boosted query:

- q: ("Christmas celebration"~2)^10 ("Christmas decoration"~2)^10 ("Christmas season"~2)^10 tradition ornamental
- q.op: OR
- qf: Introduction Description Etymology

Relevance Judgment: For this query, it was important to only retrieve plants that mention they can be used as Christmas decorations.

3.3 Evaluation

To analyze the system's performance in hand, it was a crucial step to evaluate the results obtained in the queries and compare them between schemaless, simple and boosted. Individual metrics can lead to bias or tunnel vision of the system.

The relevant documents were judged and manually retrieved based on their relevance. The results obtained for all three systems, Schemaless, Simple System and Enhanced System, were compared with the relevant documents and, at last, evaluated based on their precision.

The evaluation metrics used were:

- **Precision:** Measures the ratio of relevant documents retrieved and total documents retrieved;
- **Recall:** Measures the ratio of relevant documents retrieved and total relevant documents in the dataset;
- **Average Precision:** Measures the average of precision values obtained at different recall levels;
- **Precision@10:** Measures the ratio of relevant documents retrieved among the first 10 documents retrieved.

To evaluate the precision, it was decided to focus on the first 10 documents retrieved since the dataset is small (a few hundred rows), and the results retrieved were not that many.

The recall was estimated by comparing the set of retrieved documents with the entire collection. The plots generated show a precision-recall curve, which traces the evolution of the system, as it evaluates the results of the query.

Initially, it was observed that all the queries executed without a schema had a precision of 0, either simple or boosted. This concluded that not using a schema makes the system weaker and unable to retrieve the information a user needs.

However, the queries executed with the schema, simple and boosted, provided intriguing insights which will be analyzed in the following subsections.

3.3.1. Search Task 1 - Which plants can survive in cold environments?

Table 3 - Evaluation Metrics for Search Task 1's query

Metric	Simple	Enhanced
Average Precision	0.650000	0.966667
Precision@10	0.400000	0.500000

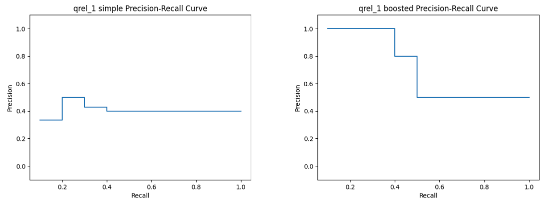


Figure 5 - Precision and Recall curves for Search Task 1's query

These results (**refer to Table 3 and Figure 5**) show that the boosted search performs better than the simple one. Both queries show almost the same precision@10 meaning the boosted search retrieved one more relevant document relative to the simple search. However, the boosted query has a greater average precision, meaning that the relevant documents it retrieved were positioned higher when compared to the simple query. This shows that the parameters of the boosted query were better tuned to find what plants could survive in a possibly cold environment.

3.3.2. Search Task 2 - Which plants have purple flowers?

Table 4 - Evaluation Metrics for Search Task 2's query

Metric	Simple	Enhanced
Average Precision	0.458333	0.794444
Precision@10	0.400000	0.500000

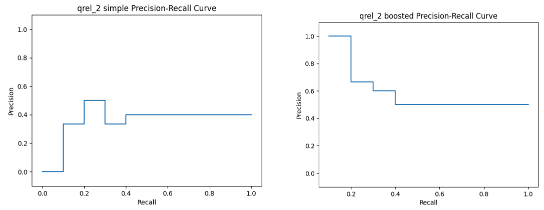


Figure 6 - Precision and Recall curves for Search Task 2's query

These results (**refer to Table 4**) show that the boosted query had an overall better performance than the simple query. It is possible to observe, from the shape of the plot (**refer to Figure 6**) curve of the simple query, that the first result it retrieves is not relevant, and immediately worsens the curve. This also shows the importance of the boosted query's weighted fields and terms when retrieving the most relevant results, namely the color purple in flowers.

3.3.3. Search Task 3 - Plants that are trees and have edible fruits?

Table 5 - Evaluation Metrics for Search Task 3's query

Metric	**Simple**	**Enhanced**
Average Precision	0.553571	0.827381
Precision@10	0.300000	0.700000

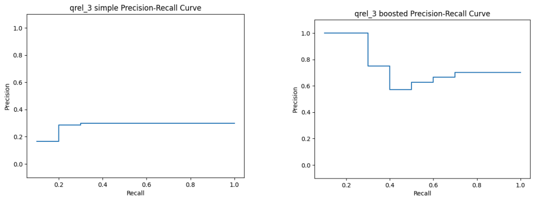


Figure 7 - Precision and Recall curves for Search Task 3's query

These results (**refer to Table 5 and Figure 7**) show that the boosted query improved the search a lot and managed to retrieve more of the relevant plants compared to the simple query by boosting the terms 'fruit' and 'trees'.

3.3.4. Search Task 4 - Which plants can be used as Christmas decorations?

Table 6 - Evaluation Metrics for Search Task 4's query

Metric	**Simple**	**Enhanced**
Average Precision	0.830357	1.000000
Precision@5	0.400000	0.300000

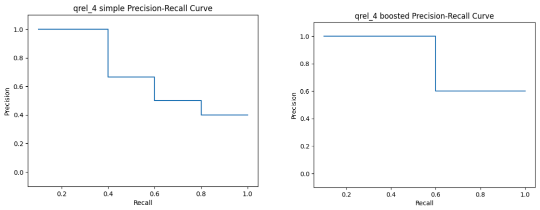


Figure 8 - Precision and Recall curves for Search Task 4's query

Although both results (**refer to Table 6 and Figure 8**) are quite good, it shows that, even in a small sample size of results, the boosted query was able to prioritize the most relevant plants associated with Christmas, when compared to its simple counterpart. Since the results returned were not many, the first 5 results were evaluated on their retrieval precision.

3.3.5. Mean Average Precision

Additionally, the Mean Average Precision (MAP) of the system, was demonstrated to be the following:

- **Simple System**: 0.62
- **Enhanced System**: 0.90

Overall, it was concluded that the enhanced system performed better than the simple one. This was to be expected, as the most relevant results are retrieved first, and are followed by less relevant results. It also highlights the impact of boosting specific parameters that match the purpose of the search task.

4. SEARCH SYSTEM IMPROVEMENTS

The third milestone is achieved by developing the final version of the search system. This version is an improvement over the previous milestone, making use of features and techniques to improve the quality of the search results. One possible enhancement thoroughly explored and evaluated involved the introduction of semantic search with the use of embeddings within the system. Additionally, minor improvements such as the identification and incorporation of more relevant results as well as refining query formulations, were implemented.

4.1 Minor System Improvements

Before applying semantic search to the system, we focused on improving the previous system by:

- **Refining query formulation**: Discarded unnecessary terms, added more query fields to retrieve more relevant results found in other fields, uniformed the term and field boosts across all queries. A simple boost has a value of 2 while a more reinforced boost has double the value, of 4. Some search task queries were reformulated to retrieve more interesting results which resulted in a slight change of its information needs.
- **Query processing**: Reevaluated the relevance feedback and identified more relevant results based on the individual assessment of reading the text of the retrieved plants for a specific query.

4.2 Semantic Search

Given the text richness of the plants' dataset, there are several situations where identical concepts are expressed in numerous variations. For example, within environmental terminology, words such as "cold" may correlate to words such as "snow", "ice", etc. Semantic search aims to understand the meaning behind a user's query, considering the intent through contextual relevance and synonym recognition, rather than just matching keywords in the text.

To implement this idea, the plants' data underwent processing via a JSON script that generated a new JSON file incorporating a "vector" field. Within this field, numerical representations of the text were stored, with the intent of encapsulating its semantic information.

Finally, a new schema was created to incorporate this new field, allowing the team to use it when querying.

This approach encompasses the following steps:

- Use a deep learning model to derive embeddings for our documents.
- Extend the existing Solr schema by incorporating a new field to store the document embeddings.
- Run queries on the new collection utilizing Solr's nearest neighbor query parser.

To retrieve documents semantically similar to a given query, the dense vector embeddings were leveraged. Due to the substantial size of these embeddings, in a Python script, POST requests were used since it is more apt for this kind of task. By running this script, simple and boosted queries were tested and the results retrieved were stored, evaluated, and compared.

4.3 Evaluation

For each search task, the evaluation of the retrieved results' precision was done differently depending on the number of results retrieved. If each system retrieved at least 10 results but less than 20, then Precision@10 would be calculated. The same logic was applied to the rest of the search tasks' queries, which use P@10 or P@20. To evaluate the overall performance of both systems, a Mean Average Precision was calculated.

The manual evaluation process for each search task was done by analyzing both results from the simple and enhanced systems for each search task. The results were analyzed in terms of relevance by looking up the terms of the query in the fields where they were looked upon and assessed on their accuracy with the expected result for the formulated query.

4.3.1 Search Task 1 - Which plants can survive in cold environments?

To majorly improve the results for this search task, in comparison to the previous milestone, more relevant results were identified. Additionally, a portion of the query was modified to enhance the precision of the resulting plants in both systems, taking a special focus on the top 20 plants retrieved. The query was refined by excluding potential irrelevant outcomes and exploring a different field that was not previously considered in the query formulation:

- **Simple query:** (cold OR "low temperature" OR "tolerate cold" OR "cold tolerate") AND NOT ("damage cold" OR "cold damage" OR "not tolerate cold" OR "not cold tolerate")
- **Simple Query Fields**: Introduction Description Characteristics Ecology Cultivation
- **Boosted query**: (cold^0.5 OR "low temperature"~1^2 OR "tolerate cold"~2^4 OR "cold tolerate"~2^4) AND NOT ("damage cold"~4 OR "cold damage"~4 OR "not tolerate cold"~4 OR "not cold tolerate"~4)
- **Boosted Query Fields**: Introduction Description^2 Characteristics Ecology Cultivation

The enhanced query enables a broader range of term matches by employing proximity search in multiple expressions. It assigns higher significance to documents containing phrases such as "tolerant to cold," "cold tolerant," and "low-temperature" compared to documents where only the term "cold" or its predefined synonyms (winter, frost, freeze, snow, or frozen) appear. This strategy acknowledges that these synonyms might be used in various contexts more frequently. After some trial and error, the approach was refined by defining weighted fields, aiming to optimize overall query performance.

Table 7 - Evaluation Metrics for improved Search Task 1's query

Metric	Simple	Enhanced
Average Precision	0.734974	0.960530
Precision@20	0.750000	0.850000
Individual Assessment	NRRNRRRRRR RRRNRNRNRR	RRRRRRRRRR RRNRNRNRRR

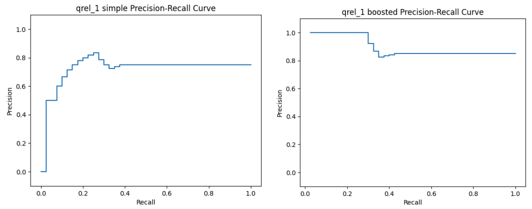


Figure 9 - Precision and Recall curves for Search Task 1's query

Each result obtained was individually assessed based on its relevance to the search task, as described in Table 8. This analysis involved a close examination of the text within each specified field. Plants recovered by the improved system that were considered irrelevant to the search included *Vaccinium ovatum*, *Hellebore* and *Cattleya schroederae*. This assessment allowed to conclude that the term "winter" is integrated with other potential plant names such as "winter rose" and "winter blueberry" or refers to the "winter dormancy" phase observed in plants.

In general, both search systems showed significant improvements by simply applying the modifications mentioned previously, leading to the retrieval of a greater number of relevant results.

Additionally, to evaluate the integration of semantic search into the system some metrics analysis in **Table 8** and **Figure 11** are provided. This resulted from examined queries which included the following:

- **Simple query**: (cold AND tolerate) OR (low AND temperature)
- **Boosted query**: (cold AND tolerate)~1^2 OR (low AND temperature)

Table 8 - Evaluation Metrics for improved Search Task 1's query with Semantic Search

Metric	Simple with Semantic Search	Enhanced with Semantic Search
Average Precision	0.162500	0.266667
Precision@20	0.100000	0.100000
Individual Assessment	NNNNNNNRNR NNNNNNNNNN	NNRNNNNNNR NNNNNNNNNN

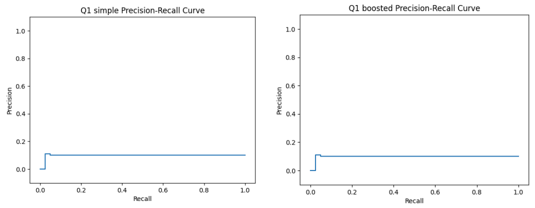


Figure 11 - Precision and Recall curves for Search Task 2's query

All the denial part of the query was removed to allow the semantic search to retrieve results simply based on the query's context. In this way, it will not look for "damage" or "not tolerate", making the system returning only plants that are related to "cold" in a beneficial way.

Despite this, the results obtained weren't very pleasant as a substantial reduction in query precision is evident, only achieving two relevant results either in the basic or in the boosted queries. It can be concluded that semantic search is not a good option in this search scenario.

4.3.2 Search Task 2 - Which plants that are not trees have purple flowers?

In the previous milestone, a lot of relevant plants were not identified and that caused the results obtained to be poor and incorrect on both simple and enhanced systems. In the third milestone, by adding more relevant results, when calculating the precision, both systems were able to retrieve relevant plants for the top 20 results, each having a precision of 1. As a way to retrieve more interesting results, it was decided to slightly modify the query and consider purple flowers in plants that are not trees. This resulted in the following query formulation:

- **Simple query:** purple flowers NOT(tree)
- **Boosted query**: "purple flowers"~5 NOT(trees) purple^2
- **Query Fields**: Introduction Description Characteristics
- **Query Operator**: AND

For this query, the search task changed to the following: "Which plants that are not trees have purple flowers?". This can be an information need for someone who has seen a purple flower in a plant that is not considered a tree, this way excluding it from the search and retrieving more precise results.

This slight reformulation provided interesting results (**refer to Table 9 and Figure 11**) and an improvement in both the simple and enhanced systems. For this search task, the top 20 results were retrieved since both systems retrieved a few dozen results and this way the systems can be compared and evaluated more efficiently on the top results.

Table 9 - Evaluation Metrics for improved Search Task 2's query

Metric	Simple	Enhanced
Average Precision	0.993808	1.000000
Precision@20	0.850000	1.000000
Individual Assessment	RRRRRRRRRR RRRRRRNNRN	RRRRRRRRRR RRRRRRRRRR

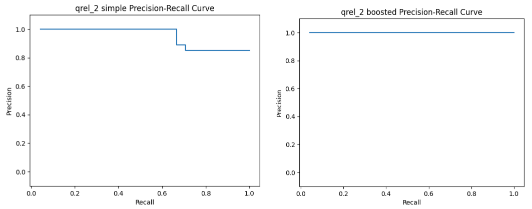


Figure 11 - Precision and Recall curves for Search Task 2's query

Each result was individually assessed in terms of relevance to the formulated query, as seen in **Table 9**, by analyzing the text in each field. The non-relevant plants the system retrieved were: *Pseudopodospermum hispanicum*, *Plantago majore*, and *Sweet potato*. In the text of the fields explored for these plants, the color purple was found to be associated with stamens, leaves, and potato skin colors. The mention of flowers was found to be completely unrelated to the color purple.

Overall, the systems showed a great improvement by reformulating queries, identifying more relevant results, and correcting the use of added boosts.

Furthermore, to evaluate the introduction of semantic search within the system, the queries evaluated were the following:

- **Simple query**: purple flowers
- **Boosted query**: “purple flowers”~5 purple^2

The term “NOT(tree)” was removed to allow the semantic search to retrieve results simply based on the context of the query. When adding the exception for the word “tree”, the system would return plants that are trees which is not what was expected.

The results obtained are presented in **Table 10** and **Figure 12**.

Table 10 - Evaluation Metrics for improved Search Task 2’s query with Semantic Search

Metric	Simple with Semantic Search	Enhanced with Semantic Search
Average Precision	0.832752	0.855269
Precision@20	0.750000	0.750000
Individual Assessment	RRRRRNNRNRR RRRRNNRRR	RRRRNRRRRRRR RRNRRRRR

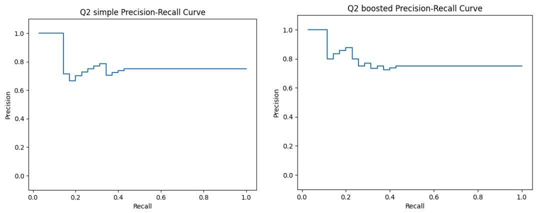


Figure 12 - Precision and Recall curves for Search Task 2’s query with Semantic Search

The results obtained from this system showed to be quite different from the system without the semantic search. The first observation was that this system returned plants such as “Orange” and “Strawberry” which are not related to the query in question and considered to not be relevant. Additionally, this system was able to return more results that were identified as relevant such as Tulip and Poppy. It is believed that these plants were retrieved based on the context of the query “purple flowers” instead of simply matching keywords like a system without semantic search would.

In conclusion, because the semantic search system returned plants that were not relevant to the search at all, it proved to worsen the search system.

4.3.3 Search Task 3 - Which plants that are trees have edible fruits that are not berries?

In order to achieve more interesting conclusions, this search task was reformulated since “plants with edible fruits” provided many positive results with little room for ambiguity. By adding an exception to the query, it was possible to test the system’s ability to filter the results of a query that obtains as many results as our previous search task.

This way the third query resulted in the following results:

- **Simple query:** fruit edible NOT(berry)
- **Boosted query**: fruit edible^4 NOT(berry)^2
- **Query Fields:** Introduction Taxonomy Description Characteristics Ecology Etymology Cultivation

Table 11 - Evaluation Metrics for improved Search Task 3’s query

Metric	Simple	Enhanced
Average Precision	1.000000	1.000000
Precision@20	0.850000	1.000000
Individual Assessment	RRRRRRRRRR RRRRRRRNNN	RRRRRRRRRR RRRRRRRRRR

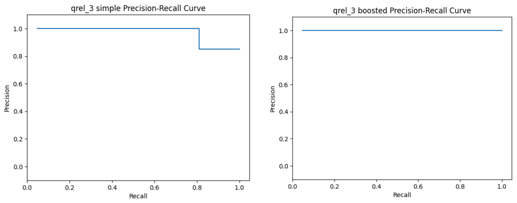


Figure 13 - Precision and Recall curves for Search Task 3’s query using old system

The new query shows good results in the improved system presenting almost only relevant results for the simple version and a totality of good results in the enhanced one. This leads us to assume this system is apt for a query like this one

Table 12 - Evaluation Metrics for improved Search Task 3’s query with Semantic Search

Metric	Simple with Semantic Search	Enhanced with Semantic Search
Average Precision	0.567073	0.615992
Precision@20	0.550000	0.550000
Individual Assessment	RNNNRNRRNR NRRRRNRRNN	NRNRNRRRRR RNRRNNNNRN

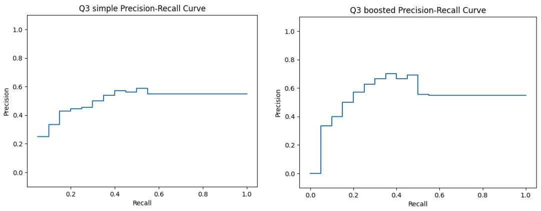


Figure 14 - Precision and Recall curves for Search Task 3’s query using semantic search

Observing these last results, it's easy to notice the dramatic difference in precision from the other system to the one using semantic search.

Evaluating the responses, we notice several plants that despite having edible fruits and being trees, their fruits are berries, leading to the thought that the semantic search does not work well for exceptions. The reason behind this would have to be investigated within the algorithm responsible for the semantic search and how it evaluates the context.

4.3.4 Search Task 4 - Which plants can be used as Christmas decorations?

In the previous milestone, the queries used for this search task were too different from the standard. As such, the queries were updated to:

- **Simple query:** Christmas decoration winter ornamental
- **Boosted query**: Christmas^4 decoration^2 winter ornamental^2
- **Query Fields**: Name Introduction Description Etymology

Both queries use the operation "OR". This is because an issue arose: many plants related to Christmas were present in the database, but were only matching the word "decoration" or "ornamental". However, using these terms in the query resulted in many other plants being retrieved, that don't have a direct relation to Christmas, but are used as decoration, regardless. Because of this, fewer arguments were used, as adding more terms or using an "AND" operator would only harm the search. The question of being more careful with the terms is valid; however, the amount of plants related to Christmas is so small that the net cast, by changing a single term, becomes either too small to detect these more hidden plants, or too wide, retrieving all decorative plants.

For this search task, the top 10 results were retrieved since both systems retrieved at least 10 results and this way the systems can be compared and evaluated more efficiently on the top results retrieved (**refer to Table 13 and Figure 15**).

Table 13 - Evaluation Metrics for improved Search Task 4's query

Metric	Simple	Enhanced
Average Precision	0.775397	0.944444
Precision@10	0.500000	0.600000
Individual Assessment	RRNRRRNNNN	RRRRRNNNRN

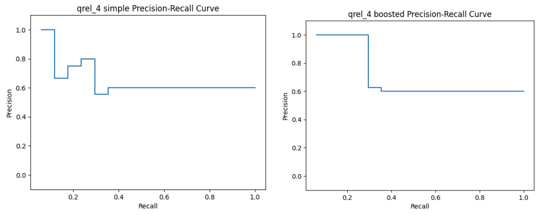


Figure 15 - Precision and Recall curves for Search Task 4's query using old system

The reformulation of the query had a slightly positive impact on the precisions and recalls of the old system. We can see in **Figure 15** that the enhanced query returns more relevant plants when compared to the simple query.

Following this, to evaluate the introduction of semantic search, the same queries were evaluated within this system. The results obtained are presented in **Table 14** and **Figure 16**.

Table 14 - Evaluation Metrics for improved Search Task 4's query with Semantic Search

Metric	Simple with Semantic Search	Enhanced with Semantic Search
Average Precision	1.000000	0.976190
Precision@10	0.600000	0.600000
Individual Assessment	RRRRRRNNNN	RRRRRRNNNN

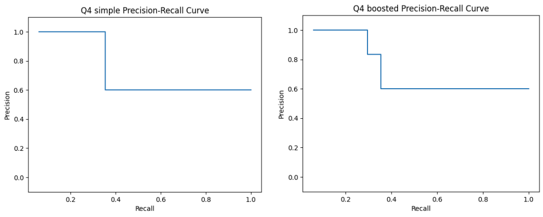


Figure 16 - Precision and Recall curves for Search Task 4's query using semantic search

With **Figure 16** we can observe a slightly worrying fact. The semantic search system has a very small impact on the performance of the query when compared to the system without semantic search. This can be a quirk of the search task query, perhaps it is too specific, such that not even using synonyms of the query terms shows a noticeable difference; Or simply further proof that the new semantic search does not result in better performance.

4.3.5. Mean Average Precision

Additionally, the Mean Average Precision (MAP) of the improved system as well as the system with the use of semantic search, in **Table 15**, were demonstrated to be the following:

Table 15 - Mean Average Precision comparison between all the systems

System	MAP
Improved Simple System	0.93
Improved Enhanced System	0.98
Simple System with Semantic Search	0.64
Enhanced System with Semantic Search	0.68

Based on the results presented in Table 15, by comparing both systems, without semantic search and with semantic search, which have a MAP of 0.95 and 0.66, respectively, the introduction of semantic search proved to worsen the system which was not the outcome initially expected.

5. CONCLUSION

This report summarizes the work done for a three-part project about Information Retrieval and Processing.

During the project, a dataset about plants was meticulously scraped, prepared and explored - for the Data Preparation phase. Next, during the Information Retrieval phase, the dataset information was prepared to be used by the retrieving information tool Solr. With this tool, it was possible to perform search tasks and improve the system using an enhanced schema and enhanced queries.

In the last phase, the team focused on improving the system previously defined by making minor modifications and corrections as well as testing the performance of the system with the introduction of semantic search.

In the end, it was concluded that the minor improvements significantly made the system perform better and as expected. The use of semantic search, however, turned out to worsen the results retrieved, which is believed to be because of the way it interprets the query. Perhaps, the diverse richness of textual data created a greater potential for misinterpreting the context of the search task.

REFERENCES

[1] Wikipedia, “Wikipedia, The Free Encyclopedia” Wikipedia, 2023. [Online]. Available: https://www.wikipedia.org/. [Accessed in September 2023].

[2] “List of plants by common name” [Online]. Available: https://en.wikipedia.org/wiki/List_of_plants_by_common_name. [Accessed in September 2023].

[3] “Welcome to Python.org” Python, [Online]. Available: https://www.python.org/. [Accessed in October 2023].

[4] “beautifulsoup4” PyPI, [Online]. Available: https://pypi.org/project/beautifulsoup4/. [Accessed in September 2023].

[5] “Project Jupyter” Jupyter, [Online]. Available: https://jupyter.org/. [Accessed in September 2023].

[6] “Asterales” Wikipedia, 18 March 2023. [Online]. Available: https://en.wikipedia.org/wiki/Asterales. [Accessed in October 2023].

[7] “Asteraceae” Wikipedia, 24 August 2023. [Online]. Available: https://en.wikipedia.org/wiki/Asteraceae. [Accessed in October 2023].

[8] “Welcome to Apache Solr - Apache Solr”, Apache Solr, [Online]. Available: https://solr.apache.org// [Accessed in October 2023]

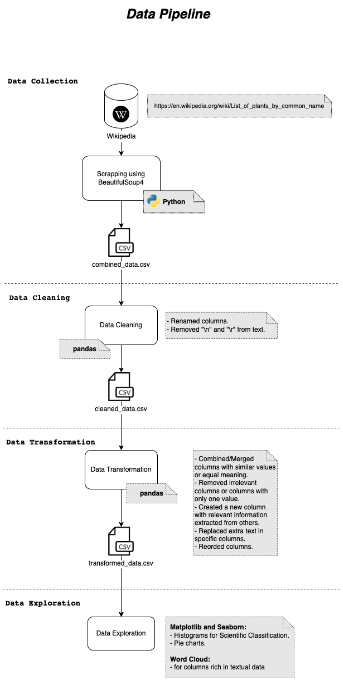
[9] “Tokenizers | Apache Solr Reference Guide 6.6”, Apache Solr, [Online]. Available: https://solr.apache.org/guide/6_6/tokenizers.html [Accessed in October 2023]

[10] “Filter Descriptions | Apache Solr Reference Guide 6.6”, Apache Solr, [Online]. Available: https://solr.apache.org/guide/6_6/filter-descriptions.html [Accessed in October 2023]

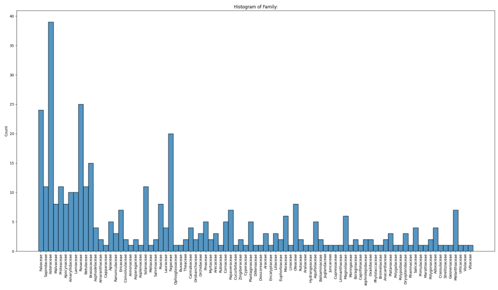
[11] “The Extended DisMax Query Parser | Apache Solr Reference Guide 6.6”, Apache Solr, [Online]. Available: https://solr.apache.org/guide/6_6/the-extended-dismax-query-parser.html [Accessed in October 2023]

A APPENDICES

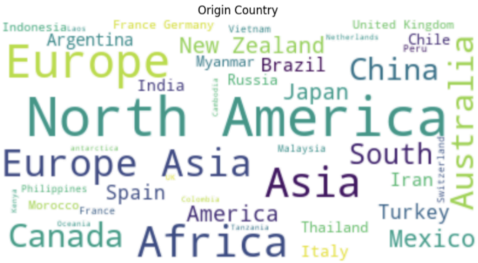
A.1 Pipeline


Data Pipeline
Data Collection
https://en.wikipedia.org/wiki/List_of_plants_by_common_name
Wikipedia
Scrapping using BeautifulSoup4
Python
CSV
combined_data.csv
Data Cleaning
Data Cleaning
- Renamed columns.
- Removed "\n" and "\r" from text.
pandas
CSV
cleaned_data.csv
Data Transformation
Data Transformation
pandas
- Combined/Merged columns with similar values or equal meaning.
- Removed irrelevant columns or columns with only one value.
- Created a new column with relevant information extracted from others.
- Replaced extra text in specific columns.
- Reorded columns.
CSV
transformed_data.csv
Data Exploration
Data Exploration
Matplotlib and Seaborn:
- Histograms for Scientific Classification.
- Pie charts.
Word Cloud:
- for columns rich in textual data

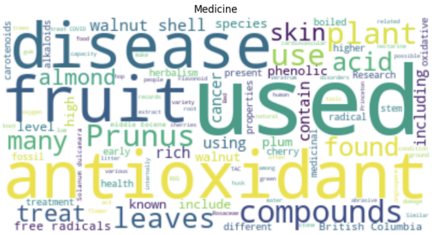
A.2 Barplot for Family



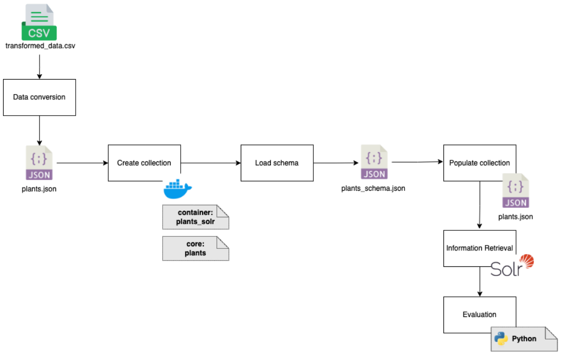
A.3 Word Cloud for the column “Origin Country”



A.4 Word Cloud for the column ‘Medicine’



A.5 Process taken of the Information Retrieval phase


CSV
transformed_data.csv
Data conversion
JSON
plants.json
Create collection
container: plants_solr
core: plants
Load schema
JSON
plants_schema.json
Populate collection
JSON
plants.json
Information Retrieval
Solr
Evaluation
Python