Fungi-On: An Information Search System for Fungi Taxonomy and Field Observations in the Iberian Peninsula

Guilherme Sequeira
up202004648@up.pt
Faculty of Engineering, University of Porto
Porto, Portugal

Pedro Nunes
up202004714@up.pt
Faculty of Engineering, University of Porto
Porto, Portugal

Pedro Ramalho
up202004715@up.pt
Faculty of Engineering, University of Porto
Porto, Portugal

ABSTRACT

This project aims to introduce an information search system on fungi, developed on top of data collected from publicly accessible sources. The development process includes preparation and processing of this data to ensure it's quality and relevance. This search system is expected to fulfill the various information needs of a broad range of end users interested in knowing more about the fungal diversity and distribution in the context of the Iberian Peninsula.

CCS CONCEPTS

• **Information systems** → **Information retrieval**.

KEYWORDS

Fungi, Fungus, Dataset, Information Processing, Information Retrieval

1 INTRODUCTION

In this study we address the intersection of information retrieval and mycology, focusing on the extensive vital class of organisms, fungi. Fungi play diverse roles in ecosystems, from supporting plant life to providing valuable pharmaceutical compounds.

To streamline data access, we leverage the Global Biodiversity Information Facility (GBIF) [**?**], an international data infrastructure committed to open access to a broad array of datasets [**?**] from multiple authoritative sources. GBIF enables precise searches and the retrieval of curated data subsets. With these resources, our goal is to develop a specialized search engine tailored to fungal species and occurrences. We concentrate our efforts on specific geographical regions, namely Portugal and Spain, to compile a comprehensive dataset. Additionally, we plan to utilize Application Programming Interfaces (APIs) for supplementary data collection.

2 MILESTONE 1 - DATA PREPARATION

The first milestone encompasses the selection, preparation, and characterization of a dataset. It results in a reproducible pipeline for data processing, the output of which is a collection of documents that will serve as a base for the subsequent milestones.

2.1 Data Processing and Preparation

This subsection outlines our systematic approach to collecting, cleansing, and structuring data on fungal species and observations. In the following subsubsections you may find a detailed explanation of each of these tasks, as well as a conceptual data model and a reproducible pipeline.

2.1.1 **Data Collection**. As previously mentioned, structured data was sourced from GBIF, a reputable international data infrastructure. Two primary datasets were acquired: `occurrences`, which contains information regarding fungal species occurrences, and `multimedia`, which provides visual representations such as images and pictures associated with these occurrences. Both datasets were available in tab-separated values format. The `occurrences` dataset comprised over 700,000 rows and had a file size of 784 MB, whilst the `multimedia` dataset contained over 120,000 rows and had a file size of 28 MB. The data obtained from GBIF belong to different datasets, each with one of three licenses: `CCO 1.0`, `CC BY-NC 4.0` and `CC BY 4.0`, all of which permit the copy and redistribute the material in any medium or format and the remix, transform, and for us to build upon the material, as long as it's for non commercial purposes as per the `CC BY-NC 4.0` license. A list of all sampled datasets is provided in the annex.

In addition to structured data, our research encompasses the retrieval of unstructured data to enrich our information repository. This data includes the collection of abstracts from scientifically relevant articles on fungal species, as well as summarized content from Wikipedia pages dedicated to individual fungal species. The extraction of this data makes use of PubMed's (the database used for the querying of scientific articles) and Wikipedia's API's. For each species, a dedicated `JSON` file is created, which aggregates the contents of abstracts with the summary of their respective Wikipedia page.

2.1.2 **Data Processing**. From the `occurrences` dataset, we extracted a comprehensive list of all observed species. This list formed the foundation for a new dataset, specifically designed to capture essential species-related data, including taxonomic information such as kingdom, family, and vernacular (common) names. A second dataset dedicated to the records of species observations was generated from the

Fungi-On, Nov 16, 2023, PRI course, M.EIC

trimmed `occurrences` dataset. To ensure data completeness and reduce redundancy, we merged features with identical semantic content. Key attributes, such as latitude, longitude, species name, and observation date, were integrated into this dataset. Finally, a third dataset was created to store visual representations associated with observations. This visual content was extracted from the `multimedia` dataset.

Following the creation of these datasets, they were subsequently loaded into a relational database. This database structure provides an efficient and structured platform for managing and querying the data.

2.1.3 **Conceptual Data Model**. The conceptual data model represents the main entities of our system: `SPECIES` (which contains species-specific information), `OBSERVATION` (which contains data about observations), `IMAGE` (which contains data about the visual representations of each image). The model is designed to provide a structured representation of the information, and `ABSTRACT` (which contains the contents of abstracts).

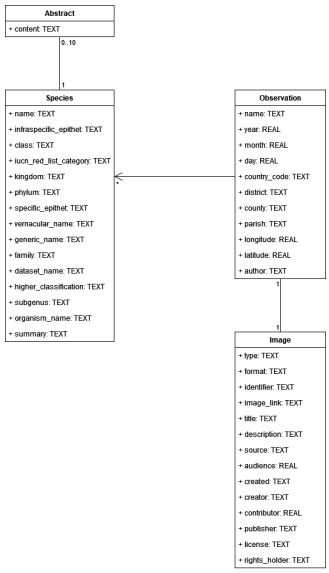

Figure 1: Conceptual Data Model.

`SPECIES` entity:

- `name`: The name of the fungal species.
- `infraspecific_epithet`: Additional taxonomic information specifying subspecies or variety.
- `class`: The taxonomic class of the species.
- `iucnRedList_category`: The IUCN Red List category indicating the species' conservation status.
- `kingdom`: The taxonomic kingdom to which the species belongs.
- `phylum`: The taxonomic phylum of the species.
- `specific_epithet`: Taxonomic information specifying the species within a genus.
- `vernacular_name`: Common or colloquial name of the species.
- `generic_name`: The generic or genus name to which the species belongs.
- `family`: The taxonomic family of the species.
- `dataset_name`: Name of the dataset from which the species data originates.
- `higher_classification`: Further taxonomic classification above the species level.
- `subgenus`: Taxonomic subgenus or subdivision.
- `organism_name`: The name of the organism or species.
- `summary`: The summary of the Wikipedia page associated with the species.

`OBSERVATIONS` entity:

- `name`: The name of the observed fungal species.
- `year`: The year of the observation.
- `month`: The month of the observation.
- `day`: The day of the observation.
- `country_code`: The country code indicating the country of observation.
- `district`: The district where the observation took place.
- `county`: The county or regional division.
- `parish`: The specific parish or locality.
- `longitude`: The geographical longitude coordinates of the observation.
- `latitude`: The geographical latitude coordinates of the observation.
- `author`: The author or observer responsible for the observation.

`IMAGES` entity:

- `type`: The type or category of the image.
- `format`: The file format or image format.
- `identifier`: An identifier for the image.
- `image_link`: The link or reference to the image file.
- `title`: The title or caption associated with the image.
- `description`: A description or additional information about the image.
- `source`: The source or origin of the image.
- `audience`: The intended audience for the image.
- `created`: The date of creation or capture of the image.
- `creator`: The creator or author of the image.
- `contributor`: Contributions or additional contributors to the image.
- `publisher`: The publisher or source responsible for publishing the image.
- `license`: The license or usage terms associated with the image.
- `rights_holder`: The rights holder or entity with rights to the image.

`ABSTRACT` entity:

- `content`: The content of the abstract.

2.1.4 **Data Pipeline**. The data pipeline serves as the structured framework for our data preparation operations. It encompasses a sequence of designed steps to transform and refine raw data pertaining to fungal species, observations and content. Each stage within the pipeline fulfills a specific

role, such as data extraction and cleansing. Conceptually, it operates as a structured workflow that systematically enhances the data's usability and accessibility. The figure below visually demonstrates the pipeline:

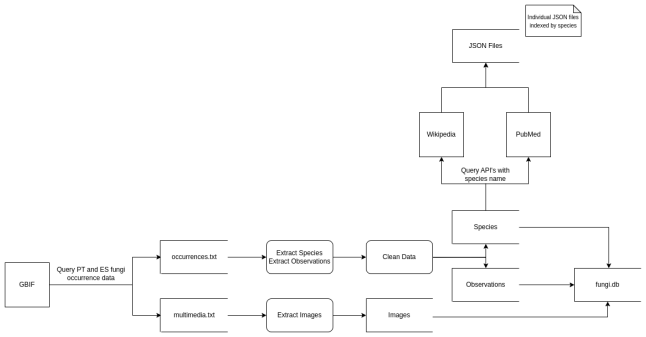

Figure 2: Data Pipeline.

To ensure that the entire data preparation process can be reliably recreated, the pipeline must be reproducible. In order to automate this process, a Makefile was designed. The Makefile enables us to automate and streamline the data preparation procedures from start to finish.

2.2 Data Characterization

An analysis was conducted in order to understand the characteristics of the collected data. In the following subsections, we present the results of this analysis as a summary of the data.

2.2.1 **Document Presentation**. In the final version of our project we aim to have two types of documents: one relative to a species and one relative to an observation.

The goal is to present the following information in the document relative to a species:

- The species' taxonomic name, including: phylum, class, order, family, genus and species.
- Its vernacular name, if one exists.
- Its edibility.
- Its toxicity to humans.
- The number of observations of the given species present in our database, as well as an address to permit access to the documents of those observations.
- A catalog of images of observations of the species.
- An interactive map that summarily displays the location of said observations as well as the rough number per region.
- A list of abstracts of related scientific publications.
- The Wikipedia summary related to the species.

As for the document relative to an observation:

- The latitude and longitude of the observation.
- The country, district, county and parish of the observation.
- The photographs associated with the observation, if present.
- The date of the observation.
- The name of the observer/rights-holder.
- The GBIF ID relative to the observation.

2.2.2 **Descriptive and Exploratory Statistics**. By plotting various graphics over the resulting data we were able to gain valuable insights on our document collection by visually representing complex information.

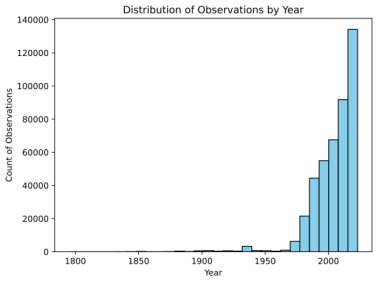

Figure 3: Observations Distribution By Year

It is noticeable that the majority of the observations were registered over the 21st century and that the number of observations has been increasing over the past few decades. This could mean an increase in data accessibility and an improvement on fungus research and registration. Overall it seems to indicate good quality data as well as a tendency for more data to be available in the future.

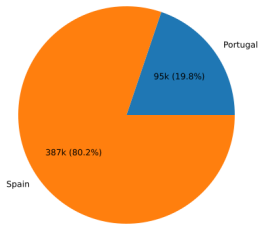

Figure 4: Observations Count By Country

The number of registers available for Portugal and Spain seems to fairly represent the difference in area between these countries, so we conclude that the data adequately represents the diversity and characteristics of both countries, minimizing any potential bias.

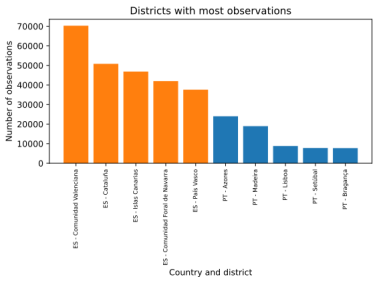

Figure 5: Observations Distribution By District

As expected, the districts with most observations in Spain have a larger number of observations than the ones from Portugal. We can also observe that the archipelagos have the most observations from Portugal.

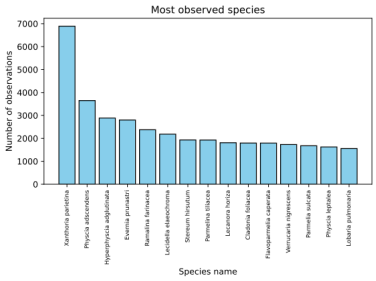

Figure 6: Most Common Species

The 10 most frequent species' number of observations are all within the same order of magnitude.

2.3 Search Scenarios

Prospective search tasks and information needs are fundamental concepts in this type of project.

Information needs refer to the underlying reasons or motivations for conducting a search. They are the goals that users are trying to achieve when using an information retrieval system. These can vary widely, and might include tasks like finding specific facts, making decision, solving problems, or simply satisfying curiosity.

As an example, consider the following information needs:

IN1 *A user wants to research the toxic effects of fungi of the Amanita genus.*

IN2 *A user is looking for scientific articles or abstracts related to diseases associated with fungi species.*

To satisfy **IN1**, the user might start by entering a query like '*toxic fungi*' onto our system. After submitting their query, the user will receive a list of search results, where they may find relevant information. There may be a scientific article which satisfies this exact need, which talks about toxic and poisonous fungi (for example, *Pholiotina rugosa*). The abstract of said article may be within the reach of our system.

3 MILESTONE 2 - INFORMATION RETRIEVAL

The second milestone focuses on the collection and indexing of data in the search engine, as well as the information retrieval process and it's subsequent evaluation. For this project, Apache Solr [**?**] was chosen as a search platform.

3.1 Documents and Data Importing

At the start of the current milestone, our data both stored in an SQLite database and JSON files. Considering that our database had a total of three tables, it was deemed appropriate to create an individual core for each entity. Additionally, it is important to note that the JSON files were storing information of up to 10 abstracts, as well as the summaries of the Wikipedia page of a species. Since these parameters differ on the type of information they provide to the user, it was also deemed appropriate to separate each one of them to their own collection, which led to the creation of two more cores. In total, we opted to define five types of documents, each one relative to a dedicated Apache Solr core:

- **Species:** species-specific information
- **Observations:** observations-specific information
- **Images:** images-specific information
- **Abstracts:** abstracts related to a given species
- **Summaries:** summaries of wikipedia pages related to a given species

In order to start working with this information in Solr, the necessary cores were created and populated with the appropriate data. Given the proportions of our system, the uploaded data consisted of a small subset of the data at our disposal. This facilitated the future evaluation of our retrieval processes. However, it is important to note that the selected sample aimed to contain entities which were the most rich in information, avoiding duplicated and missing values.

For the creation of cores, and uploading of data, we resorted to the use of Solr's REST API.

3.2 Indexing Schema

Although Solr is able to automatically perform a set of operations to identify field types in the data imported, we found it helpful to create a customized schema for our documents to have more control over the search behaviour in our search system. To upload the custom schema we resorted again to Solr's REST API. This step should be done before populating the collection, so it can be properly indexed.

We built a dedicated schema for each collection in our system. The created field types are detailed in table 1. Please see Appendix-A for the definition of each schema. However, let's discuss the intricacies behind the `longText` type. It is used for text attributes with a significant length, namely for the contents of summaries or abstracts. At index time, the analyzer splits the words into tokens, performs ASCII folding, by removing accents and diacritics, and lower cases all characters. It proceeds to apply synonym expansion to the tokens, allowing multiple tokens with similar meaning

Table 1: Schema Types

Name	Class	Tokenizer	Filters
`shortText`	`TextField`	`StandardTokenizerFactory`	`ASCIIFoldingFilterFactory`, `LowerCaseFilterFactory`
`longText`	`TextField`	`StandardTokenizerFactory`	`ASCIIFoldingFilterFactory`, `LowerCaseFilterFactory`, `SynonymGraphFilterFactory`, `FlattenGraphFilterFactory`
`mint`	`IntPointField`	*N/A*	*N/A*
`mstring`	`StrField`	*N/A*	*N/A*
`mdate`	`DateRangeField`	*N/A*	*N/A*
`mdouble`	`DoublePointField`	*N/A*	*N/A*

to be associated with the same position in the token stream, and also flattens the token graph structure, transforming the graph structure into a linear structure. Similar operations are performed at query time, so that the resulting tokens match the indexed ones.

The default Solr schema typically allows multi-value fields, resulting in single-value lists for data like simple numerical values. This doesn't seem to impact search results, however we have specified single-value types for fields where it's applicable. We have chosen not to index all fields, as some were simply not relevant for search purposes, although they're still stored and retrieved in the search results.

Below you may find a detailed description of our schema's fields:

3.3 Information Retrieval

The query system provided by `Solr` has the purpose of fulfilling the information needs presented in section 2.3. These can be translated into `eDisMax` queries. In order to demonstrate the efficacy of each boost type, we curated two distinct queries. Our selection is designed to underscore the practical utility and effectiveness in each boosting mechanism. The details of each query are documented in the following subsections.

3.3.1 **Query 1:** *A user wants to research the toxic effects of fungi of the Amanita genus.* Initially, the user would simply filter the documents whose species have the genus *Amanita*. Then, they could come up with some keywords related to their topic of search, like *toxic*, *toxicity*, or even *poisonous*. Therefore, the computed query that satisfies this need could be:

```
species: amanita*
AND
content:'toxic'^2 content:'toxicity' content:'poisonous' content:'poison'
```

The asterisk in `amanita*` is a wildcard character that represents zero or more characters, which means that species like *Amanita muscaria* and *Amanita phalloides* are included in the search. The content field looks for documents related to the keywords input by the user, namely those who match the terms *toxic*, *toxicity*, and *poisonous*, giving more weight to *toxic* due to the term boost attributed to it, specified by the `^2` after it.

3.3.2 **Query 2:** *A user is looking for scientific articles or abstracts related to diseases associated with fungi species.* The computed query that satisfies this need could simply be `content:'disease'`, which looks for documents that match the term *disease*. However, since the `content` field is equipped with a synonym filter, the retrieved documents will not only match this term, but also the defined synonyms, like *illness*, *sickness*, *disorder*, *condition*, and *malady*, which increases the overall number of relevant retrieved documents.

3.4 Evaluation

In order to evaluate the performance of our queries we compare a manually selected list of relevant documents with the ones returned by the system. To assess the improvements provided by our schema, we compared it to a simpler counterpart, using only lower case filters. The metrics utilized to compare our schema to the simpler one are a **Precision-Recall Curve**, the average precision and precision at 10, which are described below.

- Average Precision (AP): The average of the precision values obtained for the set of top N documents existing each time a new relevant document is retrieved.
- Precision at 10 (P@10): Precision at a specific cut point in the result ranked list, in this case 10, i.e. the ratio of the first 10 values that are relevant.
- Precision-Recall Curve: A graph that illustrates how precision varies according to recall, providing insights into how the system performs at varying degrees of scrutiny.

Query 1

Below are present the evaluation result's for the simpler schema:

	Metric	Value
1	Average Precision	0.817857
2	Precision at 10 (P@10)	0.6

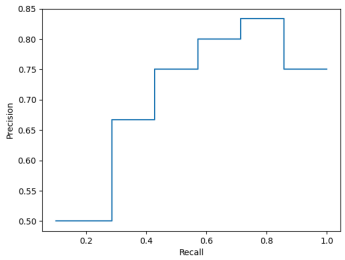

Figure 7: Simple schema's Precision-Recall Curve

As we can see below, the boosting had a noticeable effect on the performance of our system:

	Metric	Value
1	Average Precision	0.97619
2	Precision at 10 (P@10)	0.6

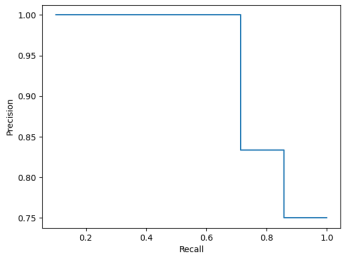

Figure 8: Refined schema's Precision-Recall Curve

Query 2

Again, below are presented the performance results for the simpler schema:

	Metric	Value
1	Average Precision	0.877381
2	Precision at 10 (P@10)	0.6

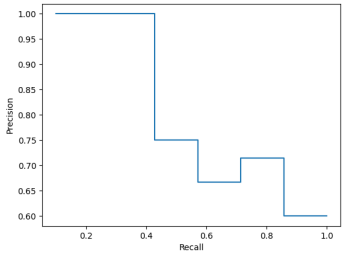

Figure 9: Simple schema's Precision-Recall Curve

Finally, the performance results for the refined schema:

	Metric	Value
1	Average Precision	0.930556
2	Precision at 10 (P@10)	0.6

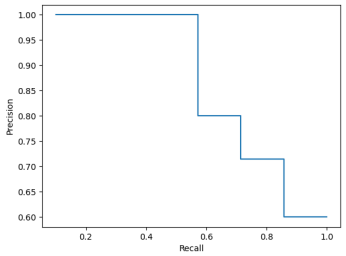

Figure 10: Refined schema's Precision-Recall Curve

4 CONCLUSION

After selecting the source and datasets to be used, cleaning and processing the data, our pipeline returns a collection of documents containing valuable information to be used in our search system. The search platform using this collection, with customized indexation and tools for information retrieval is able to return relevant information, as we've evaluated in accordance to the user's information needs. There is room for future improvement, such as refining the information retrieval process, making the search system more robust, or even broadening the search possibilities through additional data or features.

A SCHEMA DEFINITIONS

Table 2: Schema for the 'abstracts' collection

name	type	indexed
abstract_id	mint	false
species	shortText	true
content	longText	true

Table 3: Schema for the 'images' collection

name	type	indexed
index	mint	false
gbif_id	mint	true
type	mstring	false
format	mstring	true
identifier	mstring	false
image_link	mstring	false
title	shortText	true
description	shortText	true
source	shortText	true
audience	shortText	true
contributor	shortText	true
publisher	shortText	true
license	shortText	false
created	mdate	true
creator	shortText	true
rightsHolder	shortText	true

Table 4: Schema for the 'observations' collection

name	type	indexed	stored
index	mint	false	
species	shortText	true	
gbif_id	mint	true	
date	pdate	true	true
country_code	mstring	true	
district	shortText	true	
county	shortText	true	
parish	shortText	true	
longitude	mdouble	true	
latitude	mdouble	true	
author	shortText	true	

Table 5: Schema for the 'summaries' collection

name	type	indexed
summary_id	mint	false
species	shortText	true
content	longText	true

Table 6: Schema for the 'species' collection

name	type	indexed
index	mint	false
species	shortText	true
infraspecificEpithet	mstring	true
class	mstring	true
iucnRedListCategory	mstring	true
kingdom	mstring	true
phylum	mstring	true
specificEpithet	mstring	true
vernacularName	mstring	true
genericName	mstring	true
family	mstring	true
higherClassification	mstring	true
subgenus	mstring	true
organismName	mstring	true
datasetName	shortText	false