DIGIT.B4 – Big Data PoC

RTD – Health papers

D04.01.Information System

everis Spain S.L.U

Table of contents

Table of figures

1 INTRODUCTION

1.1 Context of the project

The objective of this proof of concept is to prove how big data techniques can be applied in the research domain and to demonstrate the policy benefits big data can bring.

Specifically, this proof of concept demonstrates the use of text mining techniques on large amounts of unstructured research papers as a means to identify trending topics in the health research field. This analysis can be used as an additional input prior to launching calls for grants.

1.2 Objective

The purpose of this document is to reflect the technologies used and the tool structure.

2 TECHNOLOGIES USED

In this project, multiple technologies have been used for the publication of the results. These technologies are the ones mostly used in the data analysis and visualization fields of knowledge and are widely supported by different software communities around the world.

2.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantic. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Its philosophy emphasizes code readability and its syntax allow users to express complex concepts in fewer lines of code than in other languages such as Java or C.

The Python interpreter and the extensive standard library are available in source or binary form free of charge for all major platforms, and can as well be freely distributed.

It is also a very ‘hot’ technology in the data science world, and it is widely used for the implementation of different machine learning algorithms and text mining techniques with very popular libraries such as *numpy*, *scipy* or *scikit-learn* among others.

These are the main reasons to choose this language as the base for the project.

In this project we have used many Python libraries that are discussed below.

The data analyzed in this project has been written in csv format, therefore, the *csv* library was used as it implements classes to read and write tabular data in CSV format. Additionally, programmers can also describe the CSV format understood by other applications or define their own special-purpose CSV formats.

The text had to be processed to make the classification of medical papers. To do this, the *string* module that allows manipulation of strings and the *NLTK* library were used. *NLTK* library has a collection of python packages and objects very suitable for tasks that are needed in the natural language processing.

Scikit-learn offers a series of algorithms of supervised and unsupervised learning through a consistent interface in Python. It is licensed under a simplified permissive BSD license and is distributed in many Linux distributions, encouraging academic and commercial use. The library is focused on modelling data but not on loading, manipulating and summarizing it. This library has been used to make the modeling data and identifying the category to which each paper belongs to.

Scikit-learn include modules like *SciPy* which is an essential library for scientific computing, or *NumPy,* a Python extension, which adds more support for vectors and matrices and constitutes a library of high-level math that lets the user manage those structures. These libraries have also been used.

2.2 Django

Django is a free and open source web application framework designed for python that allows rapid development and prototyping, taking care of the most common tasks performed by web developers so they can focus on writing the app instead of dealing with repetitive work.

Some companies popularly known as the Washington Post or Pinterest use Django.

In this project, Django has facilitated the use of a unique technology because the development of data modelling has been done with the Python programming language.

Django is the only framework that has the ability to generate admin panels on the fly depending on the database schema and table relations. That makes Django the most powerful framework as of today when it comes to sites where admin panels are widely used. This is the main advantage over other frameworks like PHP. Some of the benefits that Django offers over other frameworks include that the administrator is able to add new users with specific rights on the fly, as well as making user groups with rights to edit/delete content. It also has a simple but useful design (everything that any administrator will ever need is there: action history, adding/updating/deleting content, uploading images/videos/files, etc.), and better database handling to provide easy data manipulation, migration, etc.

It supports the most popular web application servers (as Apache and Nginx) and databases (MySQL, PostgreSQL).

2.3 JavaScript + D3.js

D3.js (Data-Driven Documents) is a JavaScript library that provides ways to make beautiful, dynamic, interactive data visualizations in web browsers using technologies widely implemented such as SVG, HTML5 and CSS, providing great control over the final result.

D3.js has some advantages like allowing an efficient manipulation of documents based on data. This avoids proprietary representation and affords extraordinary flexibility, exposing the full capabilities of web standards such as HTML, SVG, and CSS as explained before. With minimal overhead, D3.js is extremely fast, supporting large datasets and dynamic behaviours for interaction and animation. D3.js' functional style allows reusing code through a diverse collection of components and plugins.

One of the most important tasks of this project is the data visualization. All visualizations (bubbles, lines, bars and word cloud) were made using this library.

In the pictures below you can see all the visualizations of data which have been used with the D3.js library.

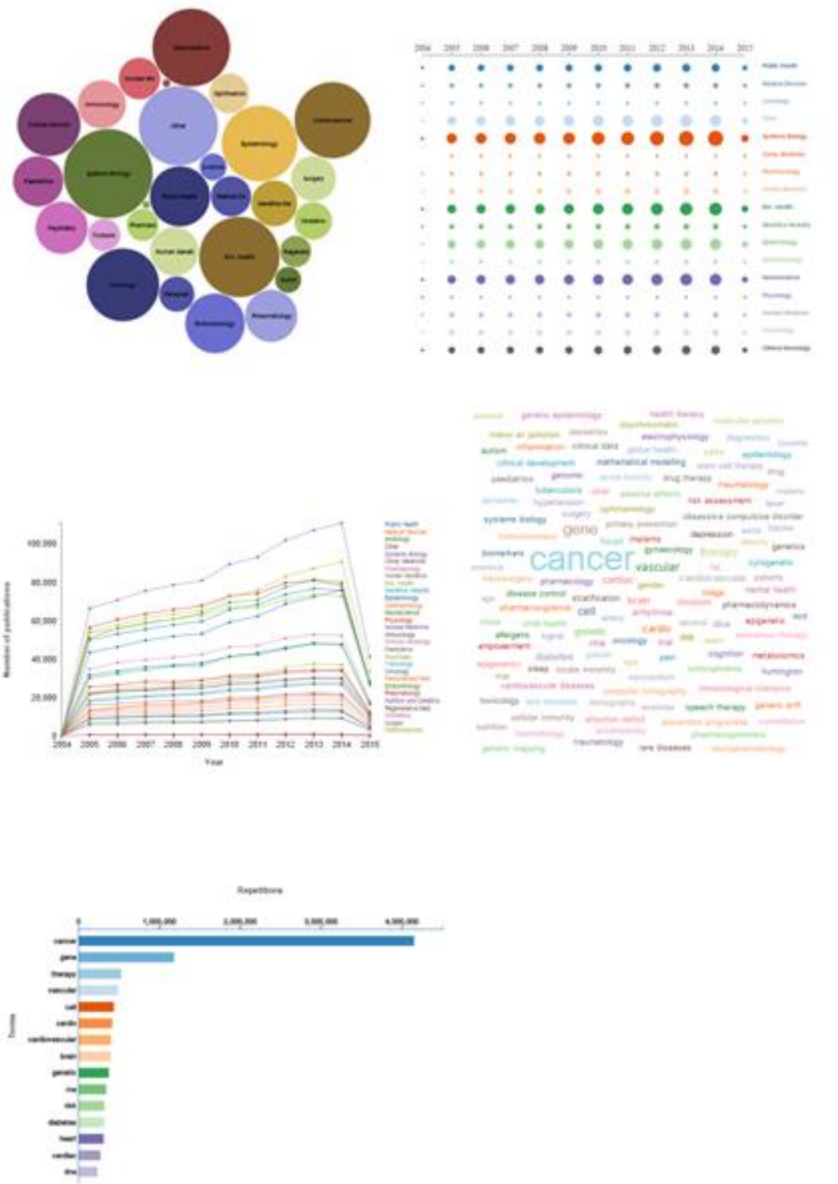


Figure 1 - D3.js example: Data visualization

2.4 HTML & CSS

HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the most popular core technologies used to build Web pages. HTML is the language that describes the structure of Web pages while CSS describes the presentation, covering the colors, layout, and fonts. It allows the developer to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup languages.

These are probably the most used web technologies, most likely for the numerous benefits they have as well as a widely spread support. HTML is an open technology, highly flexible and supported on almost every browser. CSS will automatically apply the specified styles to the elements desired. The web pages also load faster and thanks to the simplified structure and location of CSS files, changing the style of an element is much easier and faster.

These technologies have been used in conjunction with Django for the website presentation layer.

Django also provides its own template language to work with HTML that makes the work easier for developers as it makes the presentation of the application more dynamic.

2.5 Amazon EC2

One of the objectives of the project is to be as agile as possible in the creation of prototypes and versions of the application.

The Amazon EC2 (Elastic Compute Cloud) service provides a very fast and easy server deployment process, allowing the developers to ignore the details of preparing a specific machine or machines for the web application.

These web servers provide great scalability and security, and the possibility of choosing the operating system that best suits the needs of the project.

2.6 Solr

Solr is a search engine based on an open source library of the Lucene Java project with APIs for XML / HTTP and JSON. The search includes functionalities such as results highlighting, faceted search, caching, and an interface for administration.

Solr is highly reliable, scalable and fault tolerant. It provides distributed indexing, replication and load-balanced querying, automated failover and recovery, centralized configuration and many more features.

In this project it is necessary to make text searches. For this, database Mysql with full text indexes and a cluster based on HDFS (Hadoop Distributed File System) and Spark were used, but both options took too long to do the searches, for this reason Solr was used, as it optimized the searches.

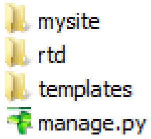
3 PROJECT STRUCTURE

3.1 Project Structure

Django offers a very intuitive structure for a project, separating each part of the project in different areas.

3.1.1 The app structure

A Django project may consist in several apps, each one with its specific directory inside the main project. In this case, only one app has been developed: the one for the visualization of the health papers analysis result, so the directory structure is as follows:

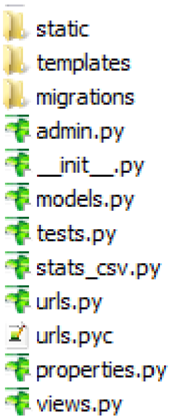


```
mysite
rtd
templates
manage.py
```

Figure 2: Project structure

Where rtd is the app, mysite is a folder that belongs to the project and the manage.py file is a Django script to run several tasks not related with the development.

The rtd app is structured as shown in the following image:



```
static
templates
migrations
admin.py
__init__.py
models.py
tests.py
stats_csv.py
urls.py
urls.pyc
properties.py
views.py
```

Figure 3: App structure

3.1.2 Application logic

The application logic can be found in the root directory of the app. Here, the views.py file is where the functionality of each web page is implemented, and where all the custom libraries (such as stats_csv.py) are located.

Here is also where the URLs for each web page are defined (urls.py), the URL for Solr (properties.py) and the models for the data (models.py and stats_csv.py).

3.1.3 The application templates

The web application uses HTML templates to show the results and the information gathered from the analysis. These templates are located in the “templates” directory, and are linked to the views.

3.1.4 The application static files

Web applications usually use some kind of static files for the presentation layer. In this case, the static folder contains:

- Images of the application (logos, icons, etc.)
- JavaScript visualizations: the JavaScript visualizations are static script files that read from a data source to show the results of the analysis.
- CSS files: the CSS style sheets are also located in this folder, as they won’t dynamically change in the application.