Art Analysis

Ana Silva, Fábio Araújo, Gonçalo Santos, Susana Lima
Faculdade de Engenharia, Universidade do Porto, Portugal
{up201604105, up201607944, up201603265, up201603634}@fe.up.pt

***Abstract*—This work focuses on collecting, cleaning, analyzing, and querying datasets concerning artworks and artists. For this, data from *SemArt*, a dataset of various artworks, is complemented with artists' data collected from *DBPedia*. This process is followed by the cleaning and refinement of the data, which is then analyzed to investigate various patterns. Through the usage of a search engine, *Solr*, a collection is defined, and the respective documents are indexed. This allows the answering of different information needs while exploring the tool's main features. The results of the queries are then assessed in order to evaluate the implemented system. Finally, an ontology representative of the domain is created and populated with Semantic Web tools. The remaining retrieval tasks are answered via *SPARQL* queries obtaining results as expected.**

***Index Terms*—Data refinement, Data analysis, Information Retrieval, Search Engines, Artworks**

I. Introduction

Art is a language valued by everyone around the globe, it can't be defined or restricted to one single type. Art should be easily accessed and available for anyone who wants to enjoy it, but this is not always the case. Despite of the fact that some platforms try to fulfill this information gap, none does it concerning global relevant artwork or in a user friendly way.

This article describes the first development phase of a novel platform, connecting art enthusiasts with artwork from all corners of the world, using an intuitive and powerful search engine, customisable through filters of different kinds, ranging from structured to unstructured data.

The remaining of this article is split in three major parts. The first is related with the data preparation process and is divided into six sections. The *Data Collection* section, describes the sources' information, the process of data gathering, data enrichment and creation of the first version of the dataset. The limitations found during this process are detailed in the *Data Limitations* section. The data preparation and refinement is stated in the fourth section. It focus on the process of normalising and cleaning the original data so later it's easier to handle it. The *Conceptual Model* section describes how the dataset will be organised and structured. It is followed by *Search Tasks*, in which all the queries that will be possible to do in the platform are stated. Finally, the characterisation of the dataset is done on Section II-F, using different charts and interpreting them in order to better understand the collected data.

The second part of the paper, *Information Retrieval*, describes the information retrieval system implemented. First a brief comparison between the two main search engines is presented in the *Information Retrieval Tool* section. Then the system's collections and documents are described, *Collection and Documents*. In the following section, *Index Processing*, the documents' indexing process is also explained, focusing on the characterization of the filters applied to the most relevant fields. Finally the system is evaluated by comparing the performance of different system configurations on various queries.

The third part, *Semantic Web*, details the implementation of an ontology representative of the dataset's domain. First, in the *Existing Ontologies* section, different artwork related ontologies are explored. In the following sections, the development and population of our own ontology are detailed. Then, in the *Queries* section, *SPARQL* is used to answer the remaining information retrieval tasks proposed. The overall experience with the Semantic Web is evaluated in the *Evaluation* section. A comparison between *Semantic Web* and *Information Retrieval* is presented as well as a final section with the practical application of the developed ontology.

In the end, final remarks and conclusions reached are presented, as well as future work that can be added to improve the developed work.

II. Dataset Preparation

The dataset for the project was collected from different data sources. Then the data was cleaned with resource to a data refinement tool and further analysed through a variety of plots. The pipeline of the entire process is portrayed on *Figure 15 in Annex A*.

A. Data Collection

In an initial stage, the data was collected from *WikiArt*. *WikiArt* is presented as a visual art encyclopedia, whose main goal is to make art from different places accessible to everyone [1]. *WikiArt* provides information about 250,000 artworks from 3,000 artists. A *JavaScript* script was implemented in order to retrieve this information through the available *API* (the free package offers 4 requests per second) [2], which was then stored in a *CSV* file. Although there are both structured and unstructured data in this dataset, after some analysis, it was noted that the latter was very scarce for the majority of artists and artworks. A possible solution for this problem consisted in complementing the data with information retrieved from other resources. This revealed to be a complex task since a lot of the featured artworks (and artists) do not have information available in other places.

After further investigation, it was decided to discard the data from *WikiArt* and use instead data from *SemArt*. This is a multi-modal dataset for semantic art understanding, which includes more than 19.000 artworks, providing, for each, a textual description and other structured attributes [3]. This data is directly downloaded from the project web page. The dataset was released under a *Creative Commons Attribution-NonCommercial 4.0* license [4]. Since this dataset is mainly used for supervised machine learning tasks, only one file (*SemArt*, a *CSV* with around 13,8 MB) is used, the remaining files and images provided were not relevant. Besides the textual description (artistic comments) for the artworks, other attributes like the technique, the size, the artist's name and school are provided.

In order to enrich this dataset, information about the artist responsible for each artwork and the techniques used is retrieved from *DBpedia*, which contains structured information extracted from the *Wikipedia* project [5]. *Python* scripts that use *SPARQL* [6] to query the *DBpedia* were implemented to obtain this additional information. For each technique its' description was obtained. Regarding the artist, its' biography, birth date, place, death date and place were retrieved. In the end of this process, two *CSV* files were created, one for the techniques (4,7 kB) and the other for the artists (1,4 MB).

It should be mentioned that the data from *DBPedia* follows the *Creative Commons Attribution-ShareAlike 3.0 License* [7] and the *GNU Free Documentation License* [8, 9].

B. Data Limitations

Some limitations regarding the dataset enrichment were encountered. The artists' names provided in the original dataset have some typos or are variances of the most known name for the artist. Moreover, there are some artworks with unknown artists.

The way an artist's information is retrieved requires the name to match the one present in *DBpedia*. For the majority of the artists (around 2,000) this was not a problem since the match was successful. For the remaining artists the *Custom Search JSON API* [10] was used to retrieve the correct name and then query the *DBpedia* for the necessary data. This is done by enabling flags that reduce the search scope to the english *Wikipedia*.

After the enrichment process, only 400 of the 3,300 artists existent in the dataset don't have any additional information.

C. Data Cleaning and Refinement

In order to better categorize the paintings, there was a field containing the techniques and the materials used that was split in *Technique*, *Material* and *Size*. Initially, as there were some paintings with more than one technique, it was considered to split the technique column in multiple ones, but as only around 400 paintings (from more than 19.000) had multiple techniques, this idea was disregarded and only the first technique was considered. In order to search for the artists on *DBpedia* and to display them correctly, it is important that the name of the artist is formatted in a certain way and capitalized, so their names were capitalized, normalized and formatted from "<LastName>, <FirstName>" to "<FirstName> <LastName>". The artists' biography and techniques' description were also cleaned, removing some irrelevant parts, such as the pronunciation of the artist name. The painting's date was also cleaned, as sometimes it had more than one date or some additional text. When more than one year appeared in that field, only the first one was considered. The techniques and materials fields were also refined, techniques/materials that were the same or considered very similar but written in a slightly different way were joined.

This process was done on *OpenRefine* [11], an open source tool to clean and transform tabular data. The results were achieved using regular expressions [12], short *Python* scripts and clustering.

D. Conceptual Model

The conceptual model consists of several classes, with the *Painting* class being the main one. The other classes help to complement it with additional relevant information. Those classes are *Artist*, *Material*, *Technique* and *School*. The main class has several attributes that result in the complete information of a painting. Namely:

- *artist* (artist name) that will link to the *Artist* class, within there is information about the artist's birth and death date, places of birth and death and the biography.
- *school* (school name) that links to the *School* class, where there is information about the school where the artist went.
- *technique* (technique name) that links with the *Technique* class, where information about the technique used in painting can be found.
- *material* (material name) that links to the *Material* class where it has information on the type of material used in the painting.
- *description* where you have some textual information about the painting.
- *date* that represents a margin of years in which the painting was performed.
- *width and height* that tells us the final dimensions that the painting has after being finished.

The conceptual model schema can be observed in Figure 1.

E. Search Tasks

With the information obtained, some of the possible queries to the database are:

- Search a painting for its name, date, artist, size, type, school, description. - Return a list of paintings filtered with the desired parameters.
- Search an artist for its name, date and place of birth and death, paintings, biography. - Return a list of artists filtered with the desired parameters.
- Check the evolution of an artist's paintings' thematics over time. - Return a list of paintings that belong to an artist ordered by years when the paintings were made.

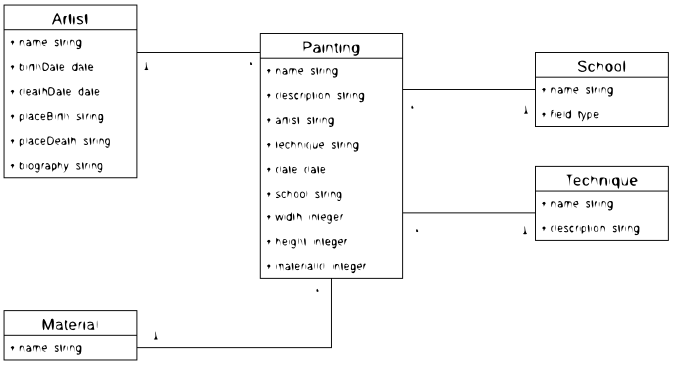

Fig. 1: Conceptual model.

This allows to see the diversification in thematics over the years of an artist.

- Check the evolution of paintings' thematics in a certain school over time. - Return a list of paintings that belong to artists who went to a school ordered by years when the paintings were made. This allows to see the diversification in thematics over the years in one school.
- Check the time when paintings of a certain type were done. - See the evolution of types of paintings over the time.
- Given an artist show others related to them by date of birth, date of death, place of birth or place of death.- Return all artists who have something in common with a given artist.
- Given a school show the artists with the most paintings.- Return a list of artists who did more paintings over time at a given school.

There are a few online encyclopedias, such as *Artcyclopedia* [13] or *WikiArt*, that enable some of these searches, such as searching a painting for its' title or an artist by name, but neither of them offers it with a good user experience or this many filters.

F. Dataset Characterisation

In order to better understand and characterise the collected data, charts were developed concerning the paintings distribution along the years, the materials and techniques used, the artists existing in the dataset and some attributes of the unstructured data.

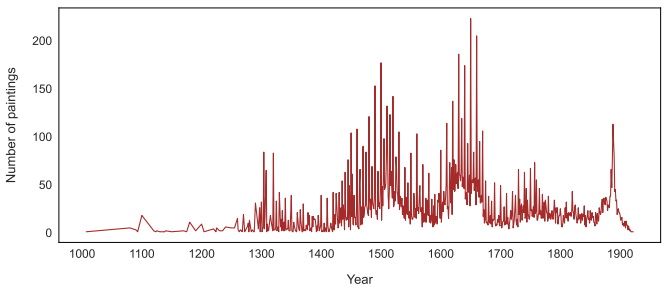

Fig. 2: Number of paintings over time.

Regarding the year a painting was started, it can be concluded checking the chart in Figure 2 that the dataset focus on paintings started between 1,400 and 1,700 and that there is a great fluctuation of the number of paintings even between two consecutive years.

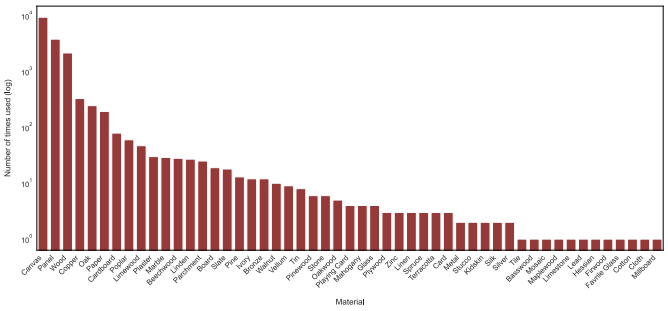

Fig. 3: Number of times a material was used (log).

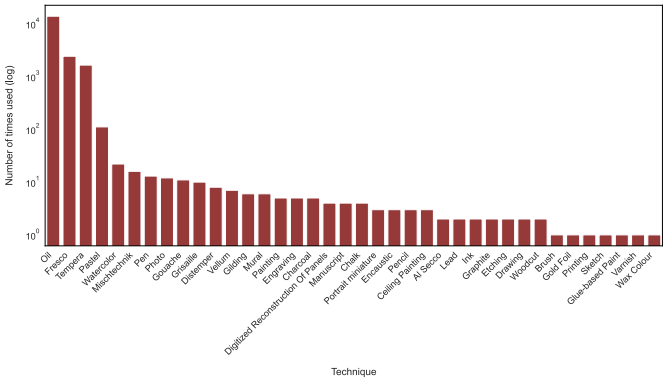

Fig. 4: Number of times a technique was used (log).

Concerning the materials and techniques used in a painting, as shown in Figures 3 and 4, although there is a big variety of both on all paintings (51 different materials and 42 different techniques) the three most frequent are used, respectively, in 93% and 97% of the paintings. Both plots used the logarithmic scale so that they can be presented in a more compact way.

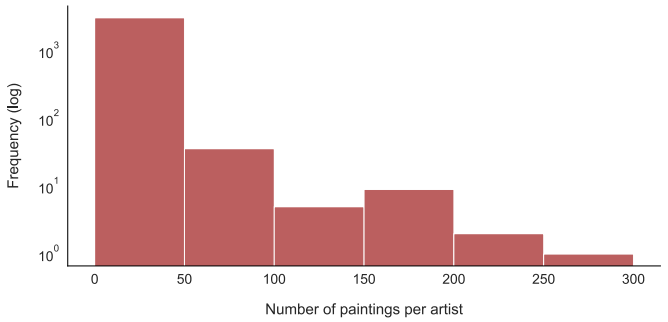

Fig. 5: Histogram of the number of paintings performed by each artist.

There are a lot of different artists present in the data set, in total 3,144 for 19,163 artworks. 98% of them have between 0

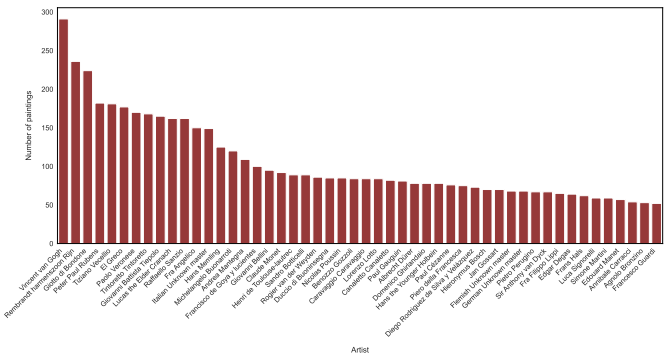

Fig. 6: Number of paintings of the top 50 artists.

and 50 artworks in which 37% only have one. An histogram of the number of artworks performed by each artist can be seen in Figure 5. As it would be expected, famous artists are among the ones that authored the most amount of art pieces in the data set. For example, Vincent van Gogh alone is responsible for 291 of them and other well known painters like Claude Monet, El Greco, Rembrandt have more than 70 each. Moreover, Vicent Van Gogh only uses 3 different materials (canvas which he used in 278 artworks, cardboard and paper). The 50 artists with the most art pieces can be found in Figure 6.

Regarding unstructured data, it was studied the number of words both in an artwork description as well as in the artist biography. Given the boxplot in Figure 7, it's possible to conclude that the descriptions have usually a bigger length than the biographies, namely the first quarter of the descriptions' length has a very similar value to the fourth quarter of the biographies' length.

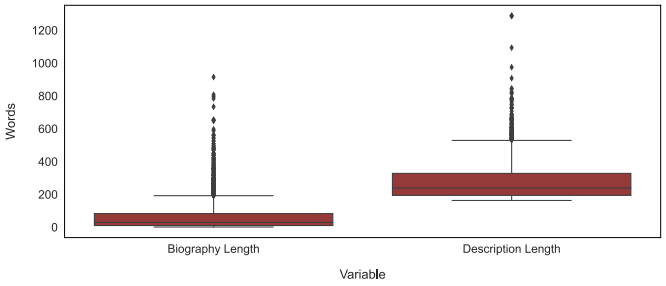

Fig. 7: Boxplot of descriptions and biographies lengths.

Finally, taken into account the Word Cloud plot in Figure 9, one of the bigger words, hence one that appears often, is renaissance. Renaissance was not only but also an artistic period taking place from the 14th to 17th century, which is close to the period of time shown in Figure 2 that has more paintings. This might be because of the large number of painters from that period existent in the database.

III. Information Retrieval

Information retrieval is the process of finding material, usually documents, within a collection that satisfies an information need [14]. This work explores *ad hoc* retrieval tasks, were the information need is specified through a user-initiated query.

Fig. 8: Descritptions' keywords word cloud.

Fig. 9: Biographies' keywords word cloud.

In order to develop a complete and efficient system an iterative approach was adopted. First the system is improved by adding custom filters to the description and title fields which allows a more dynamic and flexible search. Various filter combinations are used and the best is selected as basis for exploring different weights' configurations. These systems are evaluated by assessing the 20 first results obtained and calculating different metrics, such as the precision at k (Precision @ k), recall at k (Recall @ k), Average Precision (AvP) and Mean Average Precision (MAP).

A. Information Retrieval Tool

Two different information retrieval tools were taken into account, *ElasticSearch* [15] and *Apache Solr* [16]. Since both engines are built on top of the same core – *Apache Lucene* [17] – they support similar features, such as faceting, boosting, filters, full-text, fuzzy and proximity searches, and offer a functional and documented *REST API* [18, 19].

ElasticSearch is a more recent project, therefore, its support community is not as wide as *Solr*'s, which is more mature. Despite that, it has a very well organized and high-quality documentation. These tools also have a different main focus, *ElasticSearch*'s is on scaling, data analytics, and processing time series data to extract relevant patterns, as *Solr*'s is on enterprise-directed text searches [18, 19].

After analyzing the advantages and disadvantages of each tool, it was decided that *Solr* was the search engine that best suited the system in development. There is a lot of information available regarding the tool, it is easily customisable and it supports all the desired features. The main disadvantage of using this tool is that it does not support multiple document types per schema, but there are simple approaches that can be adopted to surpass this issue, which are explored in more detail in section III-B.

B. Documents and Collections

The main document of the system is the artwork. This document contains all the data required to describe an artwork, such as its title, author, description, and other information. Since the created dataset contains specific details on artists and techniques, documents that represent these classes were also considered.

To be able to use multiple documents, different approaches can be taken into account. The first is to use a different *Solr* core for each document type. Although these cores are independently queried, a unique list with all results can be obtained by using the *JOIN* command on the query or by joining the independent results lists (requires the normalization of the scores). With this approach, three different collections would be used.

Another possible method is to define a flexible schema that is compatible with the different types of documents and contains attributes that would be defined or not according to each type. This implies that only one collection is used, and can be achieved by specifying the necessary fields for each document and flagging them as not required.

The current collection has 19,163 entries and only considers artworks documents, hence neither of these approaches has been applied. Nevertheless, the latter is more appropriate for the implemented system. After further thought, the techniques' document was discarded since it was irrelevant for the retrieval tasks desired. Therefore, there is only one more document that should be added to the collection, which, due to its structured and consistent nature, can easily be done by altering the artwork's scheme.

C. Information Needs

To evaluate the different information retrieval systems developed, a set of information needs were created, namely:

1) Paintings influenced by Peter Paul Rubens, where the relevant paintings have been influenced by Peter Paul Rubens, a Flemish painter of the 17th century.
2) Sacred monuments, where the relevant paintings have a sacred monument in foreground, such as a mosque, church, cathedral or synagogue.
3) Portraits of Virgin Mary, where the relevant paintings depict the Virgin Mary.
4) Horse races, where paintings portraying horses racing are relevant.
5) Ball dancing, where the main focus of a relevant painting is people dancing in a ball room.

D. Index Processing

One of the most important steps in Information Retrieval is Indexing which reduces the documents to the informative terms contained in them. For the project in question, it only makes sense to add more steps to *Solr*'s indexing pipeline for the fields with unstructured data (the description of the artwork's document and the biography of the artist's document).

A new field type *custom_text* was created with the Standard Tokenizer and the filters presented below:

- *Stop Filter* which removes all words from a given stop words list.
- *Lowercase Filter* which converts all the uppercase letters in a token to the equivalent lowercase letter token.
- *English Possessive Filter* which removes singular possessives (trailing 's) from the words.
- *Porter's Stem Filter* which applies the Porter Stemming Algorithm for English removing the endings for conjugated verbs (ing, ed), among other operations.

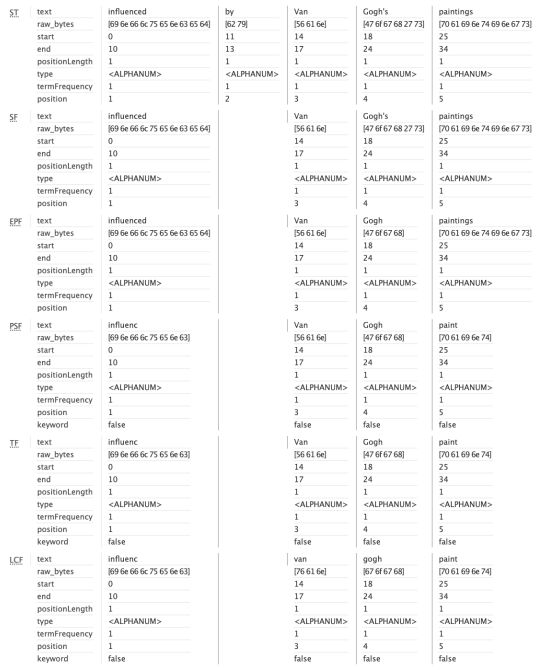

Fig. 10: An example of all the steps in index pipeline for the query "influenced by Van Gogh's paintings".

Figure 10 shows the Solr's indexing pipeline when all filters are considered for the query "influenced by Van Gogh's paintings". In it, the stopword "by" is eliminated as well as the possessive from "Van Gogh's" due to the Stop and the English Possessive Filters respectively. Furthermore, the verbs "painting" and "influenced" were reduced to "paint" and "influenc" because of the Porter's Stem Filter and at the end all the letters are lower case.

The new field type was then used for the description field of the artwork document and the biography field of the artist document.

To find the best overall system, four different Information Retrieval configurations were analyzed by incrementally adding filters to each one in order to not only find the best one but also to better understand which steps of the indexing process were more relevant. The first system considered had no filters added to the indexing process, the second one excluded a set of stop words, the third had also the Lowercase and the English Possessive Filter. Finally, the fourth one had all the previously stated filters.

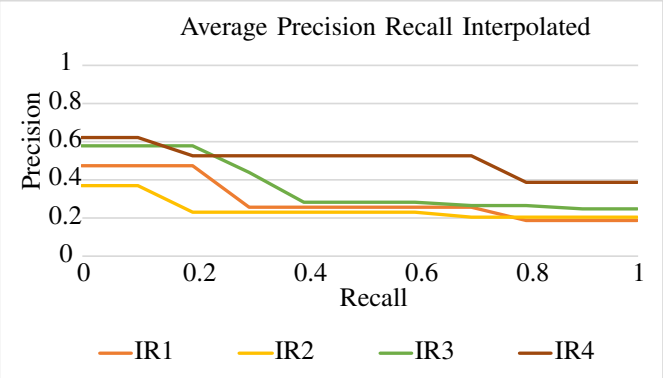

Fig. 11: Average Precision and Recall for each of the Information Retrieval Systems

As can be seen in Figures 11 none of the systems has a very good precision for the queries used. However, the fourth Information System still has a significant margin from the others. Another conclusion that can be extracted from the chart is that the exclusion of stop words in the second system doesn't improve the results. This might be due to the fact that not many stop words were used in the queries.

System	Time (s)	Size (MB)
IR1	9.5	46.01
IR2	10.1	40.88
IR3	10.4	40.44
IR4	10.6	39.72

TABLE I: Information Systems' indexing size and time.

Another important aspect that must be taken into account in order to decide the best information system is their indexing size and time. Table I shows this measurements, where time displayed is the average of five measurements. It can be concluded that there's an expected decrease of indexes size due to the fact that less words are being indexed. The greatest gap is between the first and second Information Systems given that the latter is where the stop words are firstly discarded which results in a big difference of words considered from the first to the second system. There's also an increase of indexing time due to the also increasing complexity from system to system. However, the time difference between all the Information Systems is not enough to discard one or favor another.

As such, the fourth Information System was chosen.

E. Retrieval Process

Solr offers a variety of features to enhance the retrieval process. As mentioned in Section III-A *Solr* allows the definition of custom filters for different fields in order to create more dynamic searches.

It is also possible to give weights to certain field as an effective way to boost their relevancy by setting *Boosts* to the desired field. This can be applied when the relevancy of finding a match is dependant of the field where that occurs, using the *qf* parameter. Another feature explored is the use of proximity search operators, which allows to search for terms that are within a specific distance from one another [20]. This is done by setting the *pf* attribute to a specified value corresponding to the maximum distance. Finally, while querying, different filters can be set to select which fields should be searched on.

These features were explored by answering a set of information needs and manually assessing the list of top 10 results obtained. To better understand the filters and weights role on the relevancy of the retrieved queries, three different approaches are taken into account: a base query with no enhancements, one with filters, and finally one with filters and custom weights, tailored for each information need. The information needs answered in this section are not the ones detailed in Section III-C, as they intent to enhance the differences between the three approaches, having an overall different purpose. As the goal is to explore *Solr* functionalities and their role in the relevancy of the results, no further evaluation is made in this section (see Section III-F for the systems' evaluation).

Information need: Artworks that contain at least one fisherman but do not belong to the French school.

In this information need, relevant documents portrait artworks that do not belong to the French school and contain at least one fisherman. Filtering improves the results obtained since it removes all the artworks from a french school. However, the use of boost weights achieves the best results overall (see Table II). In this case boosting the title of the artwork was enough to improve the results since it is sufficiently representative of the artwork.

query: *Fisherman*	Results	Relevant
base	23	RRRNNNNRNN
Filter: !SCHOOL:French	17	RRRRNNNNNN
weights: $TITLE^{3.0}$	17	RRRRRNNNNN

TABLE II: Query "Fisherman"

Information need: Artworks that contain a ghost, and are not religious paintings.

Artworks that contain a ghost and are not of type religious are deemed as relevant. The type of the artwork is filtered in

order to remove the religious artworks. Weights are applied in both the description and the title, giving more priority to the latter. As seen in III the third query obtains better results.

query: *Ghost*	**Results**	**Relevant**
base	47	NNRRNNNNNR
Filter: !TYPE:religious	15	RRNRNNNNRN
weights: $TITLE^{3.0}$ $DESCRIPTION^{1.0}$	15	RRRRNNNNNN

TABLE III: Query "Ghost"

Information need: Artworks of type religious or mythological that contain an idol.

Documents that contain an idol and are of type religious or mythological as deemed as relevant. In order to restrict the type of the artworks an *OR* filter can be applied to this field. Since the description is a good indicator of the content of an artwork, it can be boosted with an weight in order to improve the relevance of the results obtained. As expected, the third approach obtains more relevant results (see IV).

query: *idol*	**Results**	**Relevant**
base	41	NNNRRNRNNR
Filter: TYPE:(religious OR mythological)	33	NNRRNRNNRR
weights: $DESCRIPTION^{5.0}$	30	NRRRRRRRNR

TABLE IV: Query "idol"

F. System Evaluation

To find the best overall system, the information needs defined in Section III-C were used.

In a first stage, 4 systems were implemented only applying boost the weights. The first, *WF1*, gives priority to the title of the artwork, followed by the description and author. The second, *WF2*, prioritizes the author, then title and the description. The third, *WF3*, gives more weight to the description, then the title and author. This system was expected to perform better than the others since the description is a more reliable representation of the artwork. Finally a system with no weights, *WF4* was used as a baseline for the others. These systems were then evaluated and the best one was used as base for creating 2 other systems (the evaluation process is described in more detail in the following paragraph). These two new systems take advantage of the proximity search feature offered by *Solr*. This is done by setting the *pf* attribute to 2 in the *PWF1* system, and 5 in the *PWF2*, this proximity limit is referred as $P\backslash\boldsymbol{k}$, where **k** stands for the maximum distance between the chosen terms. The 6 boost weight factors systems (*WF*) implemented are described in Table V.

For each system, we queried it for all information needs (*IN*), compiled it and evaluated the first 20 results as *Relevant* or *Not Relevant*, as can be seen in Table XII. The

	TITLE	DESCRIPTION	AUTHOR
WF1	5	3	2
WF2	3	1	5
WF3	3	5	1
WF4	0	0	0
PWF1	5	3 + 6 P\2	1
PWF2	5	3 + 6 P\5	1

TABLE V: Boost Weight Systems.

Precision@K, Recall@K and AvP were calculated for each *IN-WF* pair, interpolated and plotted in Figure 12.

As can be observed in the Average Precision Recall Interpolated chart, *WF4* and *PWF2* were the *WFs* with best results for low recall scores, however *WF4* performance for higher recall levels was poor and similar to other systems – *WF1*, *WF2* and *WF3* – and was surpassed by *WF1*.

Analyzing the MAP@k chart, *PWF2* and *WF4* show the best results until the 10th rank, where *WF4* drops to levels similar to *WF1*. At later ranks, *PWF2* presents itself with the highest score, closely followed by *PWF1*. *WF2* is the worst system for all ranks.

Combining the evaluation of both metrics, the *PWF2* was deemed the best performer and chosen as the definitive configuration for the search system.

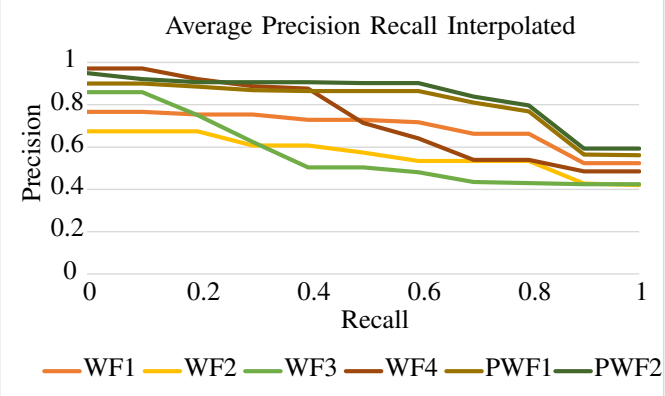

Fig. 12: Precision-Recall chart for all weight systems, for all queries average.

As expected, not all tasks had the same difficulty associated. Some of them could easily be answered by all systems – *IN5* –, while others required substantially more effort, such as *IN1*, where the *WF2* – that boosts the AUTHOR field – could not find any relevant document. Proximity search brought a great improvement in this need, as it boosted the DESCRIPTION field, where the relevant information was stated. All comparisons can be further assessed in XIII.

IV. Semantic Web

Berners-Lee et al. [21] in 2001 shared their vision on how Semantic Web could profoundly change our interaction with computers. Through a "software agent" able to consult multiple sources of information and communicate with other

agents, in the case, to arrange a medical appointment, selecting medical providers covered by the insurance plan, with an acceptable rating, not too far from its location, and not too far from home, in a simple to use and interpret manner.

This change can only happen if the information is available in a computer-readable way. Semantic Web expresses both data and logical rules on how to relate that data in the same format, using Resource Description Framework (RDF), a set of triples, similar to an elementary sentence composed by subject, verb and object. A Universal Resource Identifier (URI) identifies each element of the tuples, allowing anyone to reference an existing element or defining their own, publishing it somewhere on the Web. Using a unique URI for each concept eliminates ambiguities, that while being trivial to understand for humans – e.g. by analysing the context –, are incredibly prejudicial to computer systems. This approach creates a problem; if anyone can create a URI to define a concept, it is only a matter of time until two people create two URIs to define the same concept. Ontologies can solve this uncertainty problem. They are a collection of triples that formally defines terms' relationships. They allow computers to "understand" that two definitions may represent the same concept and introduce the concept of hierarchy, by defining classes – with specific properties – and subclasses, inheriting their parents' properties. They also provide a scheme that agents can use to navigate knowledge, following links to other ontologies.

At first, ontologies sharing our domain were explored, then the process of creating and populating our ontology was described. After having a defined and populated ontology, information needs were designed and SPARQL [6] queries were developed in order to answer them. At last, an evaluation and reflection on the experience working with Semantic Web and Protégé [22], their comparison to the previous information retrieval tools used and some practical applications that our ontology could have.

A. Existing Ontologies

We found some ontologies with some similarities with what we developed. The ontologies are:

- Towards an Ontology for Art and Colors is an ontology that refers to artwork, artist, and colors. This leaves out data such as bibliography and birth and death dates of the authors and also materials, techniques, dimensions, and school of each artwork. This ontology could be used partially in our project [23].
- Evaluation of Semantic Web Ontologies for Modeling Art Collections is an ontology already more complete and also more similar to ours, which already includes data of description, type, technique, and dimensions, which is relevant to us. But it includes data that for us are no longer needed like copyright [24].

Although we can reuse some things, we decided to make our own ontology, based on our dataset. No existing ontology covered all the data in our dataset. It allowed us to provide more complete answers. Doing this research, helped to have a better understanding of the structure of the ontologies similar to ours, aiding in the development of our own one.

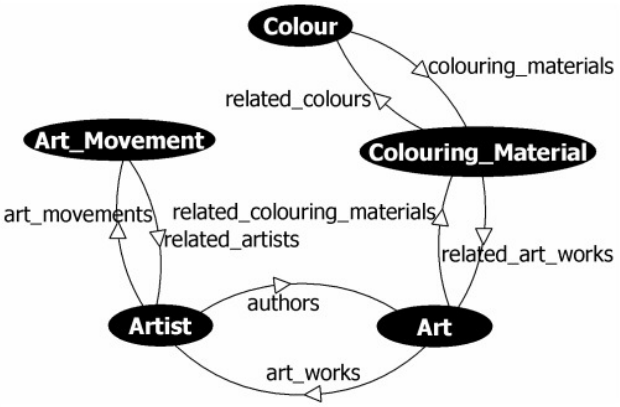

Fig. 13: Ontology Schema.

B. Ontology Creation

The artworks' ontology was created in *Protégé*.

The goal was to define a simple but complete ontology, representative of our domain. The first step to build the ontology was to define a suitable hierarchy of classes (see Figure 16), setting for each a variety of characterizing data properties (see Figure 17). The final ontology comprises 11 classes established as follows:

- **Artwork**: represents an artwork. It is characterized by 4 data properties: *Date*, *Timeframe*, *Description*, and *Title*.
- **Dimension**: represents a dimension. It comprises 2 data properties: *Unit* and *Size*.
- **Height**: subclass of *Dimension*, representing the height of an artwork.
- **Width**: subclass of *Dimension*, representing the width of an artwork..
- **Location**: represents a geographical location. It is characterized by a *Name*.
- **Material**: represents a material used to make artworks. It is characterized by a *Name*.
- **Person**: represents a person. It is characterized by 3 properties: *Name*, *DateOfBirth*, and *DateOfDeath*.
- **Artist**: represents an artist. In addition to the Person's data properties, it also includes a *Biography*.
- **School**: represents a school. It is characterized by a *Name*.
- **Technique**: represents a technique used to create artwork. It is characterized by a *Name*.
- **Type**: represents a type of artwork (such as portrait, religious, mythological). It is characterized by a *Name*.

The relations between classes can be represented through object properties. When appropriate, inverse properties were defined as well (see Figure 18). In the end 11 object properties were defined:

- **appliedOn**: relates a *Technique* to an *Artwork*. - *A technique was applied on an artwork.*
- **created**: relates an *Artist* to an *Artwork*. - *An artist created an artwork.*
- **createdBy**: relates an *Artwork* to an *Artist* (inverse of *created*). - *An artwork was created by an artist.*
- **createdWith**: relates an *Artwork* to a *Technique* (inverse of *appliedOn*). - *An artwork was created with a technique.*
- **diedOn**: relates a *Person* to a *Location*. *A person died on a location.*
- **has**: relates an *Artwork* to a *Height* or to a *Width*. *An artwork has a height/width.*
- **isFrom**: relates an *Artwork* to a *School*. *An artwork is from a school.*
- **isOfType**: relates an *Artwork* to a *Type*. *An artwork is of a certain type.*
- **usedBy**: relates a *Material* to an *Artwork*. *A material is used by at least on artwork.*
- **uses**: relates an *Artwork* to a *Material* (inverse of *usedBy*). *An artwork uses a certain material.*
- **wasBornOn**: relates a *Person* to a *Location*. *A person was born on a location.*

In order to establish a stronger connection between classes, some are also defined as subclasses of others. For example, the *Artwork* class is defined as a subclass of:

- *created by exactly 1 artist* - establishes that an artwork was created by exactly one artist.
- *createdWith some Technique* - enforces that an artwork was created with some techniques.
- *has exactly 1 height* - enforces that an artwork has exactly one height dimension.
- *isFrom exactly 1 School* - enforces that an artwork is from exactly one school.
- *isOfType exactly 1 Type* - establishes that an artwork is of exactly one type.
- *uses exactly 1 Material* - enforces that a artwork uses exactly one material.

The ontology's graph can be seen in Figure 19.

C. Ontology Population

In order to efficiently populate the ontology, the Protégé's Cellfie plugin was used. The Cellfie plugin enables the import of data from spreadsheets to OWL ontologies with transformation rules [25]. Given that our data was saved in *CSV* files it was the ideal solution and several statements were designed in the proper language, MappingMasterDSL [26].

In Figure 14 it's shown the rule to import the artworks. This rule takes advantage of several of the language's keywords, namely *Individuals*, *Types* and *Facts*. The former is used to create an individual with the name specified after the keyword. *Types* is the way to stipulate to which class that individual should belong to, which is to the Artwork's class in this case. Finally, the latter is to enumerate the individual's Sub classes and Data Properties. If a Subclass with the same name already exists, it's used, otherwise a new Subclass is created. Therefore this rule not only creates the rules, but also the Schools, Techniques, Materials, among others. Furthermore, the symbols used are also important to personalize the rule and to import efficiently. The at sign is used to reference a column in the data sheet, the asterisk makes the rule apply to each line and not to a specific cell and the text inside square and curvy brackets stipulates a regular expression to be applied to the cell's value.

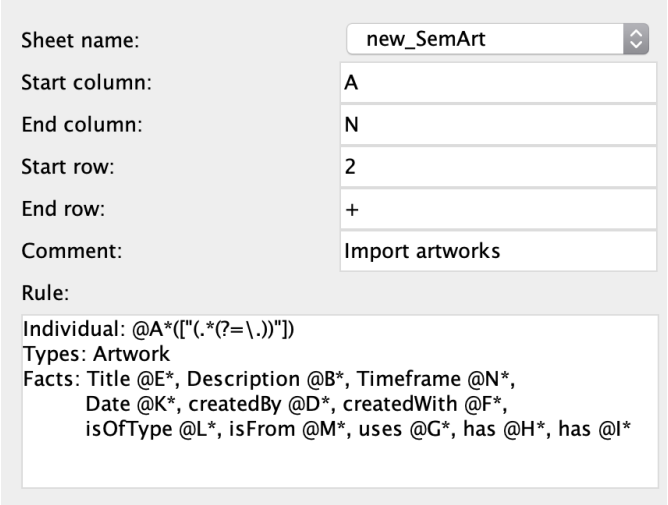

Fig. 14: Cellfie overview of the rule that imports Artworks into Protégé.

Due to Protégé's problems of handling a lot data in an efficient not all individuals were imported but a carefully chosen subset that covers most cases.

D. Queries

Several queries were developed using the SPARQL Query on Protégé to try answer the search tasks specified in Section II-E. Some of them had to be readjusted due to the version of SPARQL that Protégé implements.

It is also important to notice that a lot of results might be incomplete due the fact that it wasn't possible to import all the data into Protégé. Nevertheless, the results are shown in order to exemplify how the queries could be used and how they would satisfy the defined goals.

For all the queries shown and developed, the prefix's declaration shown in Listing 1 was used.

Listing 1: Prefixes used in SPARQL queries.

```
PREFIX rdf: <http://www.w3.org/
            1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/
            2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/
            2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/
            2001/XMLSchema#>
PREFIX : <current-ontology-path>
```

title	date	artistName	typeName
Adoration of the Magi	1481	Leonardo da Vinci	religious
Adoration of the Magi	1480	Geertgen tot Jans	religious

TABLE VI: Results for query 2.

Listing 2: Query to search an artwork based on its title.

```
SELECT ?title ?date ?artistName ?typeName ?schoolName
WHERE {
 ?artwork a      :Artwork ;
 :Title ?title ;
 :Date ?date ;
 :createdBy ?artist ;
 :isOfType ?type ;
 :isFrom ?school.
 ?artist a :Artist ;
 :Name ?artistName.
 ?type a :Type ;
 :Name ?typeName .
 ?school a :School ;
 :Name ?schoolName .
 FILTER (
  (?title = "Adoration␣of␣the␣Magi")
 )
}
```

Goal: Return a list of artworks with the desired title

A simple but interesting query is to search an artwork by a specific feature, such as its title. It allows not only to determine how many artworks exist with that feature and other relevant characteristics of the artworks, such as their artists and types. In this case, we search for artworks titled "Adoration of the Magi". The results are shown in Table VI.

This query is very straightforward since the artworks' title filters its characteristics. Other characteristics could be retrieved, such as the school and technique.

Other versions of this query can also be implemented where an artwork is chosen based on other features, such as its artist, material, type, etc... An interesting query is to select artworks based on specific keywords on their description, which can be achieved with the use of the *regex* clause.

Listing 3: Query to search an artist for its place of birth.

```
SELECT ?name
WHERE {
 ?artist a      :Artist ;
 :Name ?name ;
 :wasBornOn ?birthPlace.
 ?birthPlace a :Location ;
 :Name ?birthLocationName .
  FILTER (
        (?birthLocationName = "Leiden")
  )
}
```

name
Gerrit Dou
Geertgen tot Jans

TABLE VII: Results for query 3.

Goal: Return a list of artists born in a specific location.

Another simple query is to search artists that were born in a specific location. Similar queries could be implemented to filter artists by other characteristics such as birth date or place if desired.

In this case, we search for all artists that were born in Leiden. The results are shown in Table VII.

Listing 4: Query to understand the evolution of Leonardo da Vinci's themes over time.

```
SELECT  ?artist_name ?artwork_title
        ?type ?date
WHERE {
   ?artwork a :Artwork ;
              :Title ?artwork_title ;
              :createdBy ?artist ;
              :isOfType ?type ;
              :Date ?date .
   ?artist a :Artist ;
             :Name ?artist_name .

    FILTER
    (?artist_name = "Leonardo␣da␣Vinci")
}
ORDER BY ?date
```

Goal: To understand the evolution of an artist over time regarding their paintings' themes.

In order to better understand artist, their story and progress, it's interesting to determine which themes an artist depicts in their paintings, whether it changes along the years and if the artist went through phases, in other words, if there are periods of time when there's a focus in only one theme.

For the goal in question, the query developed selects two data properties of an artwork, its *Title* and *Date*, check its author and its theme, represented by, respectively, the *createdBy* and *isOfType* relationships. Finally, it is necessary to filter the artist to one using their name and order the results by the artwork's date so the change along the years can be depicted. Listing 4 is an example of that query in which the artist in question is Leonardo da Vinci.

As can be seen in Table VIII, in the span of 28 years Leonardo da Vinci's artworks present on this dataset focus on two main themes: religious and portraits.

Listing 5: Query to determine the most popular theme each year.

```
SELECT ?theme ?date
```

Artwork title	Theme	Date
Garment study for a seated figure	study	1470
Annunciation	religious	1472
Portrait of Ginevra de' Benci	portrait	1474
Annunciation	religious	1478
Adoration of the Magi	religious	1481
Portrait of Cecilia Gallerani	portrait	1483
Virgin of the Rocks	religious	1483
La belle Ferronnière	portrait	1490
Portrait of a Musician	portrait	1490
Madonna Litta	religious	1490
Virgin of the Rocks	religious	1495
Ceiling decoration	other	1496
The Last Supper	religious	1498

TABLE VIII: Results for query 4.

```
        (COUNT(?artwork) AS ?nr_artworks)
WHERE {
  ?artwork a :Artwork ;
  :isOfType ?theme ;
  :Date ?date .
  {
        SELECT ?date
               (MAX(?nr_artworks)
                   AS ?max_artworks)
        WHERE {
            SELECT ?theme ?date
                (COUNT(?art)
                  AS ?nr_artworks)
            WHERE {
              ?art a :Artwork ;
              :isOfType ?theme ;
              :Date ?date .
                }
                GROUP BY ?date ?theme
        }
        GROUP BY ?date
  } .
}
GROUP BY ?theme ?date ?max_artworks
HAVING (COUNT(?artwork) = ?max_artworks)
ORDER BY ?date
```

Goal: To determine the most popular theme each year.

Each artistic period is not only innovating in regards to certain techniques and materials but also to the content portrayed. For example, in the Renaissance there was great focus on mythology whereas in the Middle Age religious artworks were more common. Therefore, it's relevant to understand what was the most popular theme each year and if it's a recurring theme, or a trait of a certain era.

The goal in question, attained by the query in Listing 5, demands two subqueries, one to compute the number of artworks that depicted of theme in each year and the other to determine the theme with the most amount of artworks. Finally, the query selects the theme whose number of artworks equals the maximum amount. Therefore, all themes are returned when the number of art pieces associated to each are equal. As in the previous query, the results are ordered by date so that the evolution can be seen.

As can be seen in Table IX, in this dataset, in the span of almost 200 years most paintings depicted religious themes.

Theme	Year	Number of artworks
religious	1335	1
religious	1434	1
religious	1460	1
study	1470	1
religious	1472	1
portrait	1474	1
other	1474	1
religious	1478	1
religious	1480	1
religious	1481	1
religious	1483	1
portrait	1483	1
portrait	1490	2
religious	1495	1
other	1496	2
religious	1498	4
religious	1501	1

TABLE IX: Results for query 5.

Listing 6: Query to, given an artist, show others related to them by birth date, birth place, death date, or death place.

```
SELECT  ?nameRA
WHERE {
  ?relatedArtist  a :Artist ;
  :DateOfDeath ?dateOfDeathRA ;
  :DateOfBirth ?dateOfBirthRA;
  :Name ?nameRA;
  :diedOn ?deathPlaceRA;
  :wasBornOn ?birthPlaceRA.
   {
         SELECT ?name ?dateOfDeath
                ?dateOfBirth ?deathPlace
                ?birthPlace
         WHERE {
           ?art  a :Artist ;
           :DateOfDeath ?dateOfDeath;
           :DateOfBirth ?dateOfBirth;
           :Name ?name ;
           :diedOn ?deathPlace ;
           :wasBornOn ?birthPlace.
           FILTER (
            ?name = "Maso di Banco"
           )
          }
        }
        FILTER(
```

```
          (?nameRA != ?name) &&
          (?deathPlaceRA = ?deathPlace ||
           ?birthPlaceRA = ?birthPlace ||
           ?dateOfBirthRA = ?dateOfBirth ||
           ?dateOfDeathRA = ?dateOfDeath )
          )
}
```

Name of Related Artists
Bartolomé Carducho
Giovanni Bilivert

TABLE X: Results for query 6.

Goal: Return all artists who have something in common with a given artist.

With this query it is possible to retrieve artists that relate with a specific artist. Two artists are related if they share at least one of: death place, birth place, date of birth, or date of death. The results are shown in Table X.

To accomplish this, a subquery retrieves all the properties of the specified artist (in this case "Maso di Banco") and the main query selects all other artists that match at least one of those properties. No priority or ordering is given to the artists.

Instead of comparing the exact date of death or birth it would be more interesting to compare the years or even the days of these dates, but that would require to split the dates into substrings which as not possible in Protégé (the dates are strings since some are *unknown*). For that reason this query does not return a lot of results.

Listing 7: Query to determine the artist with the most artworks each year.

```
SELECT ?artist_name ?date
       (COUNT(?artwork) AS ?nr_artworks)
WHERE {
  ?artist a :Artist ; :Name ?artist_name .
  ?artwork a :Artwork ; :Date ?date ;
  :createdBy ?artist .
  {
    SELECT ?date
           (MAX(?nr_artworks)
               AS ?max_artworks)
    WHERE {
          SELECT ?author ?date
                 (COUNT(?art)
                     AS ?nr_artworks)
          WHERE {
                 ?author a :Artist .
                 ?art a :Artwork ;
                 :Date ?date ;
                 :createdBy ?author .
          }
          GROUP BY ?date ?author
    }
    GROUP BY ?date
  } .
}
GROUP BY ?artist_name ?date ?max_artworks
HAVING (COUNT(?artwork) = ?max_artworks)
ORDER BY ?date
```

Goal: To determine the artist with the most paintings each year.

Another relevant information that can be achieved with the data is the artist with the most artworks each year. With it, one can infer, for example, whether or not the most popular artists have more pieces.

The goal in question, attained by the query in Listing 7, demands two subqueries, one to compute the number of artworks created by an artist and the other to determine the artist that authored the most amount of artworks. Finally, the query selects the artist whose number of created artworks equals the maximum amount. Therefore, all artists are returned when the number of art pieces each created are equal. The results are once again ordered by date so that an evolution can be seen.

As can be seen in Table XI, in this dataset, Leonardo da Vinci authored the most paintings from 1498 to 1513.

Artist's Name	Year	Number of artworks
Leonardo da Vinci	1498	4
Leonardo da Vinci	1501	1
Leonardo da Vinci	1503	2
Leonardo da Vinci	1505	1
Leonardo da Vinci	1508	2
Leonardo da Vinci	1510	3
Leonardo da Vinci	1513	1
Hans the Younger Holbein	1526	1
Leonardo da Vinci	1530	1
Tiziano Vecellio	1548	1
Peeter Baltens	1560	1
Bartolomé Carducho	1595	1
El Greco	1608	1
Giovanni Bilivert	1629	1
Gerrit Dou	1647	1
Francesco Guardi	1770	1
Ary Scheffer	1835	1
James Tissot	1883	1
Konstantin Alekseyevich Korovin	1906	1

TABLE XI: Partial results for query 7.

E. Evaluation

The overall experience with Semantic Web technologies and Protégé was positive since it allowed a better understanding of the Semantic Web's use to represent different domains and link various data sources. The ontologies can be queried in *SPARQL*, a query language that resembles *SQL* in how

the queries are structured and how to access different elements. This was advantageous since most of the group is well familiarised with the latter, allowing an easy learning process. Not all queries implemented had the same difficulty level, some were more straightforward than others. The ones that required multiple subqueries and *GROUP BY* clauses were more challenging, but overall this process was not very complex or time-consuming.

Regarding the use of Protégé as a tool for building and querying the ontology, the experience was not as positive, leaving some mixed-feelings. On the bright side, the tool is well documented (to some degree), which was very helpful in the first stage of the ontology's development and population. It also offers a variety of plugins for population and visualization. Although its interface is not the most intuitive, this issue was easily overcome after further exploring and reading.

Some problems were encountered while using the tool, mainly in the querying stage. Although Protégé implements *SPARQL*, allowing the execution of queries in this language, the version it supports is not compliant with the latest available *SPARQL* version [27]. This was a problem in some queries where the use of certain functions (such as *substring* to divide strings int smaller ones) was required. This was not an issue easy to understand at first sight. While composing the queries it was not uncommon to search for specific *SPARQL* functionalities and examples to use as a guide, but some would not work on Protégé. An error would appear in this situation, but no further explanation was provided. To work around this situation, the goals of some queries were slightly altered. Protégé is also very CPU consuming. This would often cause the program to slow down and interfere with other tasks being executed.

To make the development of the ontology more collaborative, Protégé Web was also experimented with. After some exploring, it was clear this tool was very restricted in functionalities compared to the desktop version.

In the end, the ontology was successfully built, and the proposed queries (with some adaptations) were answered as expected.

F. Semantic Web vs Information Retrieval

Semantic Web and Information Retrieval are areas that handle and select data in order to satisfy a user's demand for information. However, the methods they use to try to fulfill the user's needs are different. Semantic Web is defined by the World Wide Web Consortium as Web of Data [28], that can be queried using SPARQL. It requires the user to know exactly what is looking for, to develop a query that links the data with the adequate prefixes, to be aware of the available relations and how to chain them. The result and its accuracy its solely dependent on how good a query was built and the order in which the results appear doesn't reflect the system's quality. For example, in Query 7 the order is not relevant to assess the quality of the system, but it is set by the query. A solution is always an exact match because the system doesn't try to search for similar solutions and guess what the user might want. In contrast, Information Retrieval does exactly that. The returned data is not necessarily what the user typed but something that might be relevant to the user. It analyzes words' synonyms, lemmas, importance in a sentence, among other things in order to retrieve everything that could be relevant. They are the ideal system when the user doesn't know exactly what is looking for because an exact match is not required. Moreover, the order in which the results appear is also a way to assess the quality of the system.

Given its several differences, it's obvious that each will be more appropriate than the other for different queries. On one hand, Semantic Web can easily retrieve complex queries, like artists related to a given one (as stated in Query 6) or queries that demand grouping, as in the artist with the most paintings, Query 5 or even order the results in such a way that an evolution can be better understood, Query 4. On the other hand, Information Retrieval is more appropriate to look for a specific element in an artwork, being an object, building or even a person. Moreover, subtleties can better be found in the artworks' descriptions, namely, if they are influenced by a given artist or if the artist was another disciple, among others.

G. Practical Applications

As the existing data on the internet is always growing, when we want to search for artworks, we almost always cannot find all the data we want in one search, which leads us to have to search in several sources. Thus, this ontology allows the aggregation of the most important data concerning both artworks and artists. This prototype could have applications in all areas of artworks research, allowing to retrieve information on museum sites, art sites, and even universities within this context. It can also be used as a knowledge base for researchers or even ordinary users on the internet.

V. Conclusions

All proposed goals were accomplished. Regarding the Data Preparation step, there is a better understanding of the chosen domain, the already existing data and data sets in it and which ones are relevant for this purpose. In the Information Retrieval phase, a search engine was successfully used to define a collection with artwork related documents, which was then queried and evaluated with various systems and search strategies. This allowed a better comprehension of the retrieval, indexing and querying processes. Finally, in the Semantic Web stage, existing ontologies for the art domain were explored, but as they did not answer all necessary needs, a new ontology was designed and implemented. This ontology was then used to answer all remaining information needs with success. There were a few challenges associated with the software used, Protégé, but all were successfully overcome. SPARQL has a similar syntax to SQL, which aided our transition to it and enabled us to use it effectively. Through this work, we improved our understanding on the difference between semantic web and information retrieval. While the former is more appropriated to look for exact

matches on structured data, the latter thrives on unstructured data searching.

As future work, integrating this proposed ontology with state-of-the-art ontologies is an interesting approach to enrich it.

REFERENCES

[1] *WikiArt*. https://www.wikiart.org/en/about. (Accessed on 24/10/2020).

[2] *WikiArt API*. https://www.wikiart.org/en/App/GetApi. (Accessed on 27/10/2020).

[3] Noa Garcia and George Vogiatzis. "How to Read Paintings: Semantic Art Understanding with Multi-Modal Retrieval". In: *Proceedings of the European Conference in Computer Vision Workshops* (2018).

[4] *Creative Commons Attribution-NonCommercial 4.0 License*. https://creativecommons.org/licenses/by-nc/4.0/. (Accessed on 03/12/2020).

[5] *DBpedia*. https://en.wikipedia.org/wiki/DBpedia. (Accessed on 27/10/2020).

[6] *SPARQL Query Language for RDF*. https://www.w3.org/TR/rdf-sparql-query/. (Accessed on 27/10/2020).

[7] *Creative Commons Attribution-ShareAlike 3.0 License*. https://creativecommons.org/licenses/by-sa/3.0/. (Accessed on 27/10/2020).

[8] *GNU Free Documentation License*. https://www.gnu.org/licenses/fdl-1.3.html. (Accessed on 27/10/2020).

[9] *DBpedia*. https://wiki.dbpedia.org/about. (Accessed on 28/10/2020).

[10] *Custom Search JSON API*. https://developers.google.com/custom-search/v1/overview. (Accessed on 27/10/2020).

[11] *OpenRefine*. https://openrefine.org/. (Accessed on 27/10/2020).

[12] *Information Technology Laboratory Glossary: Regular Expression*. https://csrc.nist.gov/glossary/term/Regular_Expression. (Accessed on 27/10/2020).

[13] *Art cyclopedia*. http://www.artcyclopedia.com/. (Accessed on 2/11/2020).

[14] Christopher D. Manning, Prabhakar Raghavan, and Schütze Hinrich. *Introduction to information retrieval*. Cambridge University Press, 2008.

[15] *Elasticsearch: The Official Distributed Search & Analytics Engine*. https://www.elastic.co/elasticsearch/. (Accessed on 26/11/2020).

[16] *Solr is the popular, blazing-fast, open source enterprise search platform built on Apache Lucene™*. https://lucene.apache.org/solr/. (Accessed on 26/11/2020).

[17] *Welcome to Apache Lucene*. https://lucene.apache.org/. (Accessed on 26/11/2020).

[18] Asaf Yigal. *Solr vs. Elasticsearch: Who's The Leading Open Source Search Engine?* https://logz.io/blog/solr-vs-elasticsearch/. (Accessed on 26/11/2020). Aug. 2020.

[19] Kelvin Tan. *Apache Solr vs Elasticsearch*. https://solr-vs-elasticsearch.com/. (Accessed on 26/11/2020).

[20] *The Standard Query Parser*. https://lucene.apache.org/solr/guide/6_6/the-standard-query-parser.html. (Accessed on 3/12/2020).

[21] TIM BERNERS-LEE, JAMES HENDLER, and ORA LASSILA. "THE SEMANTIC WEB". In: *Scientific American* 284.5 (2001), pp. 34–43. ISSN: 00368733, 19467087. URL: http://www.jstor.org/stable/26059207.

[22] Mark A. Musen and Protégé Team. "The Protégé Project: A Look Back and a Look Forward". eng. In: *AI matters* 1.4 (June 2015). PMC4883684[pmcid], pp. 4–12. ISSN: 2372-3483. DOI: 10.1145/2757001.2757003. URL: https://doi.org/10.1145/2757001.2757003.

[23] Luciana Bordoni and Tiziana Mazzoli. *Towards an Ontology for Art and Colours*. (Accessed on 12/10/2020).

[24] Antonis Bikakis Danfeng Liu and Andreas Vlachidis. *Evaluation of Semantic Web Ontologies for Modelling Art Collections*. (Accessed on 12/10/2020).

[25] *The Cellfie plugin*. https://github.com/protegeproject/cellfie-plugin. (Accessed on 17/12/2020).

[26] *The Mapping Master Domain Specific language*. https://github.com/protegeproject/mapping-master/wiki/MappingMasterDSL. (Accessed on 17/12/2020).

[27] *SPARQL Query*. https://protegewiki.stanford.edu/wiki/SPARQL_Query. (Accessed on 02/01/2021).

[28] *Linked Data*. https://www.w3.org/standards/semanticweb/data. Accessed: 2021-01-06.

APPENDIX

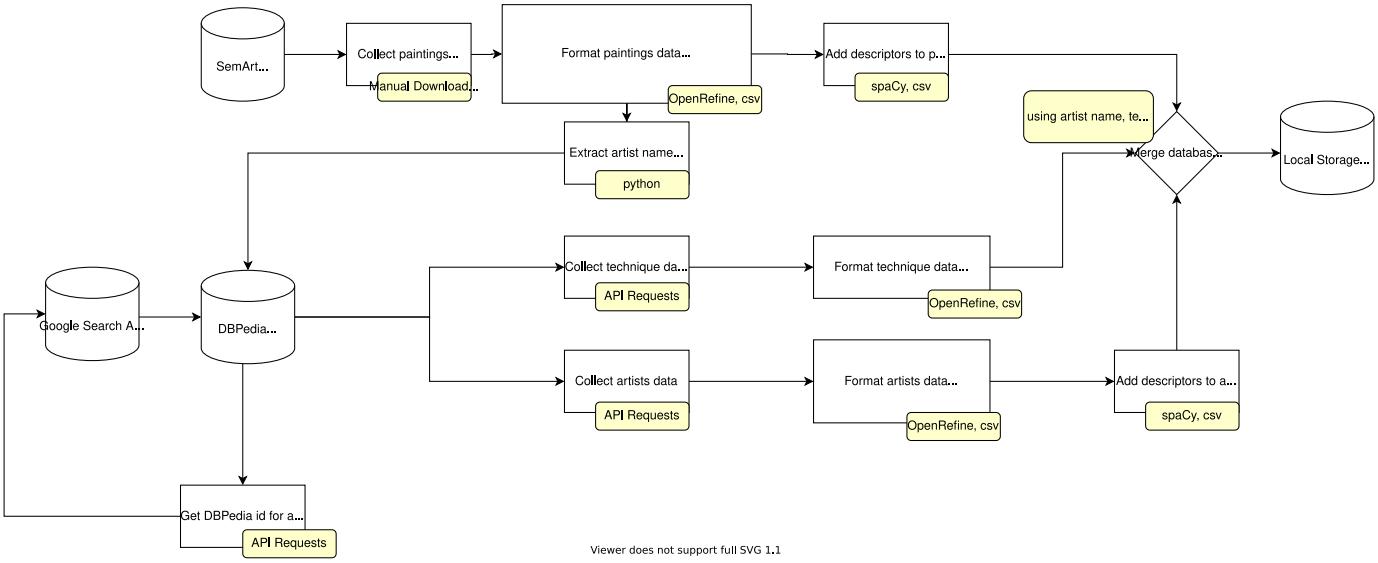

Fig. 15: Workflow Pipeline.

	WF1	WF2
IN 1	N N R N N N N N N N N N N N N N N N	N N N N N N N N N N N N N N N N N N
IN 2	R R R R R R N N N R R N N N N R N N N N	N N N R R R R R R R R R N R N N N N N
IN 3	R R N R R R R R R R R R R R R N N N N	R R N N R R N N N N N N N R R R R R N R
IN 4	N R N R N N N R N N N N R N N N N R R N	N R N R R N N N N R N N N R R N N N N N
IN 5	R R R R R R N N N N N N N N R N N N N N	R R R R R R N N N N N N N N R N N N N N

	WF3	WF4
IN 1	N N N N N R R N N R N N N R N N N R N N	R R N R R R N N N R R R N N R N N N N N
IN 2	R N N N R N N N N N N N N N R N N N N N	R R N N N N N N N N R N N N N N N R R N
IN 3	R R R N N N R R R R N N R N R N R R R R	N R R R R R R N R N R R R N R R R R R R
IN 4	R N N N R N N R R N R N N R N R N N R N	R N R R N R N N N N R N R N R N N N N N
IN 5	R N R N R N R N N N N N N N N N N N N N	R R R N N N N R N N N N N N N N R N N N

	PWF1	PWF2
IN 1	N R R R N R N R N R R N N R R R R N R N	R N R R R R R N R R R R N R N R R R R N
IN 2	R R R R R R N N N R R N N N N R N N N N	R R R R R R N N N R R N N N N R N N N N
IN 3	R R N R R R R R R R R R R N N R N R R	R R N R R R R R R R R R R N N R N R R
IN 4	N R N R R R R R N N N N R N N N N R N N	N R N R R R R R N N N N R N N N N R N N
IN 5	R R R R R R N N N N N N N N N N N N R N	R R R R R R N N N N N N N N N N N N R N

TABLE XII: Query results for all *IN-WF* pairs.

	WF1	WF2	WF3	WF4	PWF1	PWF2
IN 1	0.33	0	0.26	0.77	0.63	0.80
IN 2	0.89	0.61	0.53	0.55	0.89	0.89
IN 3	0.90	0.59	0.71	0.75	0.87	0.87
IN 4	0.38	0.46	0.50	0.63	0.59	0.59
IN 5	0.92	0.92	0.71	0.76	0.91	0.91
MAP	0.68	0.52	0.54	0.69	0.78	0.81

TABLE XIII: AvP for all *IN-WF* pairs.

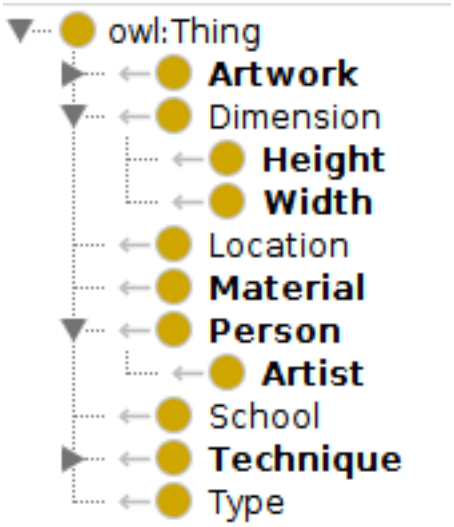

Fig. 16: Ontology classes.

Data Property	Func	Domain	Range
owl:topDataProperty			
Date		Artwork	xsd:integer
Description		Artwork	xsd:string
Timeframe		Artwork	xsd:string
DateOfBirth		Person	xsd:string
Unit		Dimension	xsd:string
Size		Dimension	xsd:double
Title		Artwork	xsd:string
Name		Technique, School, Material, Person, Location, Type	xsd:string
Biography		Artist	xsd:string
DateOfDeath		Person	xsd:string

Fig. 17: Data properties matrix.

Object Property	Func	Sym	Inv Func	Trans	ASym	Refl	Irrefl	Domain	Range	Inverse
owl:topObjectProperty								Material	Artwork	
appliedOn								Technique	Artwork	createdWith
isOfType								Artwork	Type	
created								Artist	Artwork	createdBy
createdWith								Artwork	Technique	appliedOn
has								Artwork	Width, Height	
isFrom								Artwork	School	
createdBy								Artwork	Artist	created
uses								Artwork	Material	usedBy
diedOn								Person	Location	
wasBornOn								Person	Location	
usedBy										uses

Fig. 18: Object properties matrix.

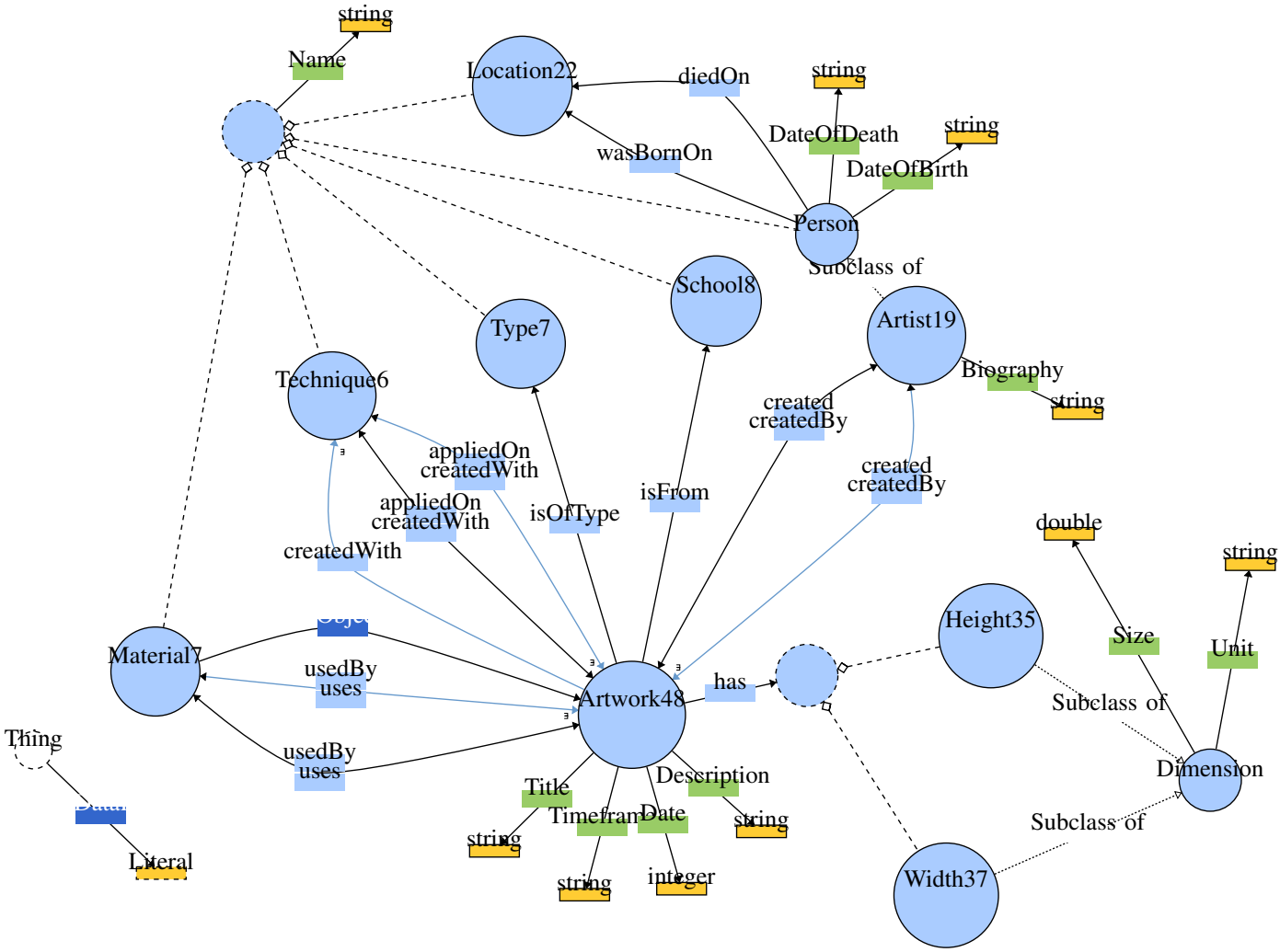

Fig. 19: Ontology graph.