Linked Product Data

Rita Curado
rita.curado@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

May 2016

Abstract

Over the years, the Web has become a place accessible to everyone and that fact was responsible for the increasing amount of data that is available today. The AAA slogan which means that "Anyone can say Anything about Any topic" describes the open nature of the Web. In this so-called "Web of Data" the task of search is becoming harder. When someone performs a search in an engine, like Google, several web pages are returned, that are related to the search by the words given by the user, but most of those web pages are documents without a well defined data schema. This makes it difficult to reference information and to compare it accurately when it comes from different sources. The aim of this work was to develop an identity mapping engine for informational entities based on Semantic Web Technologies that can find similarities between data that is kept in different data sources and presented in different web sites, and decide which of these sources are referencing the same real world objects. With that information it is possible to create links between the different data sources and offer richer information to the users, by merging disjoint information. An application for mapping and merging product data in the Web was developed and evaluated, in different scenarios.
Keywords: Linked data, identity mapping, Semantic Web, RDF, SPARQL.

1 Introduction

In the old days, when someone wanted to know something about a certain topic s/he had to search for information in an encyclopedia or go to a library and search for it in specific books. Nowadays the information is electronic and spread all over the world, accessible everywhere and kept in several formats which is collectively known as the Web. However, despite all of its advantages, the Web has become a confusing place because there are many sources of information about the same real objects but with inconsistent values for some properties. Therefore, it is on the user's hands to perform the search, decide which sources are reliable and understand which of those are referencing the same real object. This burden should be relieved.

2 Motivation

Search engines crawl the Web starting from some specific URL (Uniform Resource Locator) trying to find, in the content of those references, the list of words given by the user. Then they return a list of web pages, indexed by ID, where the number of occurrences of the given words is shown to be higher. After that, these engines give back to the user a list of web pages sorted by word occurrence ranking [4]. Therefore, the user is required to visit all the sites returned (or a reasonable number of them) to collect and filter the information s/he is searching for, i.e., he has to decide which are the web pages that reference the intended product.

3 Challenges

Search engines are optimized for quick answers but not for providing complete and unified information about a topic. To achieve this it is necessary to merge data, which, at first, might not seem a very difficult task, however the sources of information differ in their structure, therefore it is essential to standardize the data before mapping and merging it. Beyond that, inside each data source there are duplicated information which needs to be initially filtered, to not hamper the identity, mapping and merging processes.

3.1 Identity Problem

Data is kept in different databases created by different people with different points of view and different interpretations about things. This is why, to describe the same smartphone's model, certain databases will have the attribute "Name" and others will have the attribute "Model", although both attributes give the same information. The same applies to the identification of products where, according to a specific database, the product has a specific identifier that is unique inside that scope. However, many databases could describe the ex-

actly same product with different identifiers, which might mislead us to think that they are representing different objects.

In the scope of the Semantic Web, entities have a unique identifier, given by an URI, that is different according to the source they belong to, thus there had to be a way to find if two entities were referring the same real object based on their attributes.

3.2 Mapping Problem

Taking into account the impossibility of mapping entities based on its URI, another process would have to be created for us to conclude similarity based on the entities' properties. The Ontology Integration Process [12] is one process made for the purpose of merging similar information on the Web. This process is responsible for, firstly, mapping similar ontologies and then merging the information contained in each one of them. However, in our work we decided not to use ontologies, so this process had to be implemented in the scope of entities (objects references). Moreover, it had to be responsible for identifying similar entities and merging their information, thus creating a new representation of a real product, that was a mixture of several representations (entities). Figure 1 shows the goal of this work.

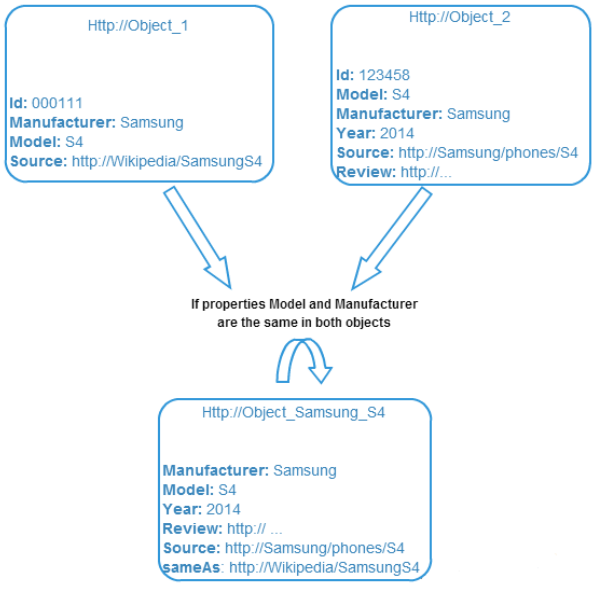


Figure 1: Linkage of information about the same product from different sources

3.3 Duplicated Data Problem

While identifying which different entities refer to the same real objects, we became aware that in the scope of products, duplicated data exists. To better explain this issue we will give an example. Imagine that a user is searching for a specific smartphone, for example the iPhone 6. In fact there will be many data sources with information about this product but in each one of them we could find variations of it, i.e., the iPhone 6 16GB, the iPhone 6 32GB, etc. Obviously, from the perspective of bar code identifiers these products are not the same but from a user search perspective the relevant information are the product characteristics and probably the variations it has. As a matter of fact, both iPhones have the exactly same properties except their storage capacity and the associated price. Therefore these two entries, from one of the available data sources, could be seen, or not, as the same real object depending on the user search's goals.

4 Use Case Scenarios

For this work we have decided to develop the application in two different scenarios, which are **Pharmaceutical Products** and **Cinematographic Works**.

We based our core implementation on the **Pharmaceutical Products Scenario** because medicines are well known products in all developed societies in the world. However, each country has its own drug regulator entity that defines which medicines can be commercialized in that same country and under which brands. Consequently, sometimes it is necessary to merge information between data available in each country and data available on the Web of Data to enrich information. An example of this kind of data available on the Web is the Drug Bank, a dataset that is part of the LOD (Linked Open Data) project and that contains loads of specific information about drugs [16].

To show that our solution was able to work in different domains, we decided to take advantage of structured data available on the web, via the Linked Open Data community and after a research for available repositories, we decided to cross information between two well known repositories: "LinkedMDB" and "DBpedia". The LinkedMDB (Linked Movie Data Base) project provides a high quality source of structured data about movies [7], while the DBpedia focuses on the task of converting Wikipedia content into structured knowledge, such that Semantic Web techniques can be employed against it (asking sophisticated queries to Wikipedia, linking it to other datasets on the Web, or creating new applications) [2]. Overall, what we would like to offer to the user is an unified answer to all the possible questions s/he can has in the pharmaceutical or cinematography contexts.

5 Contributions

This work had the aim of gathering information about products over the Web and existent repositories in the LOD community, filtering that information to remove duplicate entries, standardize it via the Semantic Web technology called RDF (Resource

Description Framework) and finally mapping and merging it to be presented to the user. Therefore, the main goal of this work was to solve the identity problem in the Semantic Web scope, i.e., identify which information was referring to the same product even though the identifiers do not coincide with each other. For that reason we developed a system, available in a public repository [1], where we were able to link information from different sources, join it into one model and offer it to the user.

The system proved to be useful and friendly in a end-user perspective, and domain independent. Besides that, for a know set of information, the system has shown the ability to find and manage duplicate information with high levels of precision, depending on the filtering rules applied, and also the ability to merge information from different data sources with high precision and recall values.

Therefore as a second goal we hoped that the reduction of the users workload could be seen as an advantage, despite of not being able to return the information as fast as the actual search engines, which proved to be achieved taking into account the feedback given to us by users who have tested the system, shown in Section 8.

6 Related Work

With the growth of data in the Web, the problem of verifying if two entities from different sources are referring the same real world object has been recognized as an important issue by several other researchers. In fact, unique identifiers are absent in most of the existing data over the Web and therefore the need of identifying real world objects became a study area. Four different approaches arose in order to solve the identity mapping problem: manual, deterministic, heuristic and probabilistic.

6.1 Manual Approach

In this approach humans take a central role in the mapping by introducing their knowledge into the system to define a mapping as a "good" or "bad" one.

Some of the works done in this area asked for user feedback in a mapping classification perspective, i.e. the users have to classify mapping rules already made by examining which columns from different tables are aimed to be joined. This type of approach was used by Alexe et al. [1], McCann et al. [11] and Yan et al. [17].

Another way of using user feedback is by showing the obtained results, by applying a mapping to a dataset, to the user and ask her/him if the returned tuples were expected or not. Belhajjame et al. [3] and Cao et al. [5] used this approach to learn from users feedback and improve the previously made mapping rules.

[1] https://github.com/inesc-id/AdvancedSearchApp

6.2 Deterministic Approach

In deterministic approaches the outcome is precisely determined through known relationships among different sources, without any room for random variation.

Raimond et al. [13] developed an automatic interlinking for music databases. They manage to find similarities in pairs of resources, from the different sources, by comparing the literals attached to them.

The Ramezani et al. [15] aimed to find association rules in Linked Data. For that purpose they collected the desired data, placed the data in a central database with a unified ontology and designed an algorithm [14] to create the association rules.

6.3 Heuristic Approach

Heuristic approaches aim to find an identity mapping which is not guaranteed to be optimal, but good enough for a given set of goals.

Isele et al. [9] developed an algorithm called GenLink that learns linkage rules from a set of reference links using genetic programming, i.e. the distance is measured to define how similar two properties are, given a specific threshold.

Friedrich et al. [6] present a totally different approach, instead of regular expressions or ontology classification, they analyze the HTML headers of the Web pages and the anchors that point to it and count specific words occurrence. Then they used HAC (Hierarchical Agglomerative Classification) as a classification algorithm in order to group objects by proximity.

6.4 Probabilistic Approach

Probabilistic approaches enable variation and uncertainty to be quantified, mainly by using distributions instead of fixed values. In the scope of link products, this approach is used to define classes of products and determine how probable a product belong to a certain class.

Kopcke et al. [10] use this approach to determine if two products' references are identifying the same real world product, thus Naive Bayes Classifier was used to calculate the probability of a product belong to a certain category. Finally, they used an heuristic approach to decide if two products refer to the same real object based on equal properties' values.

Our work combines both manual and deterministic approaches and was made in the scope of the Semantic Web using technologies such as: Resource Description Framework (RDF) a standard model for data on the Web representing data as triples which can be uniquely identified by URIs and that

can also be represented as a graph, making the linking of information much easier; SPARQL (SPARQL Protocol And RDF Query Language) a query language and protocol designed specially for querying RDF data; and OWL (Web Ontology Language) an ontology designed for the use of applications that need to process the content of information instead of just presenting it.

7 Solution

As a useful application of an identity mapping engine, we developed a linked data search application which will not be able to give a quick answer, as the current search engines, but will produce a complete and unified result. Figure 2 shows an overview of the implemented solution's architecture which is divided into four distinct modules: Data Converters; Semantic Web Engine; Search User Interface; and Data Curator Interface.

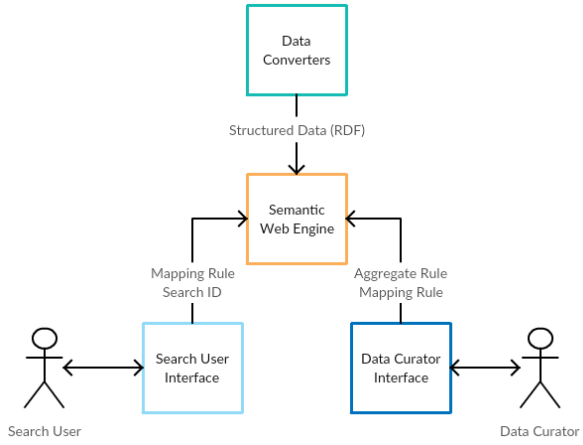


Figure 2: Solution's Overview

Our system is also divided into four main steps. First, the information is extracted from repositories on the Web and structured with the RDF framework. Second, the data is filtered to remove duplicate information that might exist in each one of the found repositories. Third, the repositories are merged by their common properties and finally, searches could be made against the system. For each of these steps there are a module responsible for dealing with the task.

The **Data Converters** (green box) module in this architecture is responsible for extracting the information and give it a standard structure to allow its manipulation. It is composed by as many Converter Processes as Data Sources, once the properties used to describe the products are not the same for every source.

The **Semantic Web Engine** (orange box) module, shown in detail in Figure 3, is responsible for the creation of filtering and mapping rules and for applying them to the main repository. Therefore, it is composed by three different repositories and by three engines.

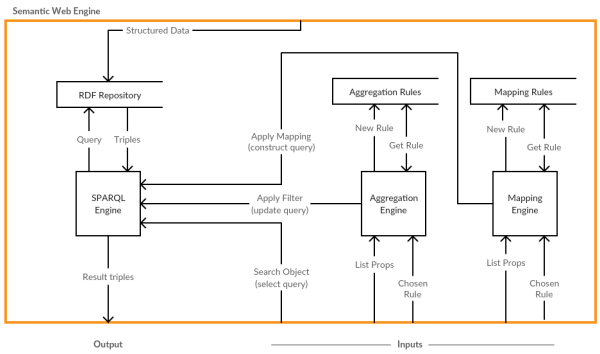


Figure 3: Semantic Web Engine

The two remaining models, **Data Curator Interface** (dark blue box) and **Search User Interface** (light blue box) are responsible for turning the interaction between users and the system possible. The first one, allows an expert user (called Data Curator) to examine each data source's properties and to choose sets of them to create both filtering and mapping rules, to be applied later for the **Semantic Web Engine**. Whereas, the second one allows common users to interact with the system by choosing which mapping rule better suits her/his needs and performing searches.

The largest effort of this work was done in the **Semantic Web Engine** module since it is the application center. This module is then responsible for all the interactions between the other modules and the main repository. It is also responsible for converting all the inputs given by both the Search User and the Data Curator to SPARQL queries, and returning the results to be then treated and presented by each of the user interfaces.

In the first part of this work, where the **Data Converters** module has to extract information from available data sources or web pages on the Web, whenever the source is a web page it uses HtmlUnit to extract the information and RDF to create a new schema for describing it. On the other hand, if the source of information is available as a "Linked Data" repository, this engine uses SPARQL to query the source and get the schema that will be needed further in the application.

For the purpose of creating filtering rules, the **Semantic Web Engine** receives a list of properties as an input from the **Data Curator Interface**, and stores that information in an Aggregation Rules repository inside it. Whenever a filtering rule is chosen, the Semantic Web Engine has to ask the Aggregation Rules repository for the list of properties associated to that rule and has to filter the RDF Repository in two steps: first, the engine as to

perform a SPARQL SELECT query to get all the entities that present the same values for the list of properties associated to the chosen rule and then, it has to perform a SPARQL DELETE query to update the repository and maintain only one entity per each pair of duplicated entities.

The task of creating and applying mapping rules represents a much more difficult task for the engine. For the purpose of creating the query, the Semantic Web Engine receives a list of properties as input and creates a SPARQL CONSTRUCT query, that is associated to the rule name, and then saved into the Mapping Rules repository. When a user (Data Curator or Search User) chooses a specific mapping rule, the Semantic Engine is then responsible for first, creating a new source model corresponding to the mapped sources; second, getting the schemas of each data source that is going to be mapped and including them in the new model; third, it has to run the CONSTRUCT query against the RDF Repository to get the resultant tuples; and finally, it has to delete from each of the mapped sources, the products that have mapped with each other (to avoid duplicates in case of searching in all the available sources).

Whenever a user performs a search in the system, a list of properties and associated values are send to the Semantic Web Engine as input, which in its turn creates a SPARQL SELECT query and apply it to the RDF repository, sending the results for the interface that asked for them.

8 Results

This work was evaluated with two approaches: an evaluation based on metrics and another based on tests with real users. With this evaluation we wanted to show, not only, the system performance on identifying duplicate data and merging data from disparate data sources depending on the chosen rules, but also its usefulness for common users.

We tested the system in the Pharmaceutical Products context and the obtained results are shown throughout this section.

8.1 Filtering Data

Since each database could have duplicate information and in this specific case of Infarmed we found some duplicate instances, we first decided to filter each data source based on their specific properties.

8.1.1 Infarmed

The data source named "Infarmed" is composed by 21 properties, 11 of which belong to the package leaflet assuming large blocks of text. In that way, from the 10 remaining properties we only use the first 5 to define duplicate information.

The 10 properties have the following names: Nome do Medicamento; Substância Ativa; Dosagem; Forma Farmacêutica; Genérico; Tamanho da Embalagem; CNPEM (Código Nacional para a Prescrição Eletrónica de Medicamentos); Preço; FI (Folheto Informativo); RCM (Resumo das Características do Medicamento). In the same order, they will be represented in the rules as follows: (N); (S); (D); (F); (G); (T); (C); (P).

That being said, we assumed that for a product to be considered a copy of another it must have the same values as the other for all the first 5 properties listed before. That way, we became aware of how many duplicate products existed in the "Infarmed" database and the ability to evaluate our own filtering rules as "Data Curators".

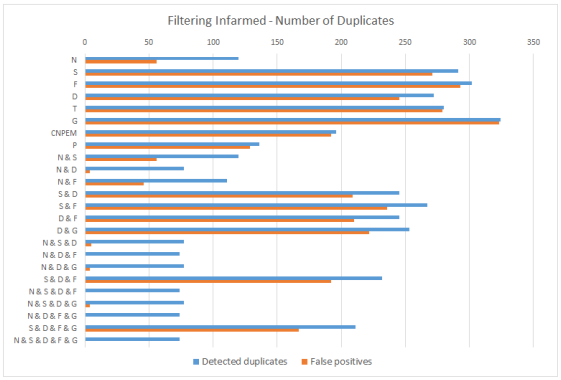
We have evaluated 33 aggregation rules, in terms of number of duplicates, true positives, false positives, precision and recall, which differ from each other in terms of complexity (number of properties tested together) and variety (for the same number of properties tested together, there are compositions of different properties). A smaller set of the evaluated rules is shown in Figure 4.

We concluded that the for the combination **Nome do Medicamento & Dosagem & Forma Farmacêutica** we could achieve more accurate results. Also, we have found that the property **Substância Ativa** produces worse results than **Nome do Medicamento** when combined with others, and that there is no rule without the property **Nome do Medicamento** that can reach the right number of existing duplicates. Regarding the number of found duplicates and the number of false positives, these two metrics reveal a directly proportional relation. In terms of precision and recall we have concluded that as the number of duplicates increases, the precision decreases, while as the number of false positives comes close to zero, the recall comes close to one. Moreover, the greater the number of properties included in a rule, the greater the accuracy of that rule.

8.1.2 Infomed

The data source named "Infomed" is composed by 5 properties, and we use all of them for the filtering process as a way of finding duplicates. Those properties have the following names: Nome do Medicamento; Nome Genérico; Dosagem; Genérico; and Titular. In the same order, they will be represented in the rules as follows: (N); (NG); (D); (G); (T).

We assumed that for a product to be considered a copy of another it must have the same values as the other for all the 5 properties listed above. That way, as in the first source we acquired the knowledge of how many duplicate products existed in the database and again the ability to evaluate our own filtering rules.



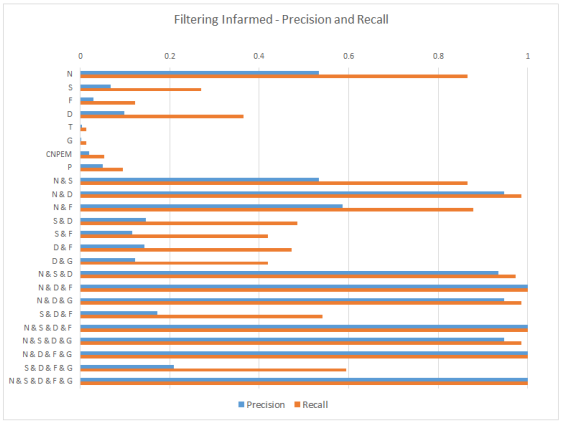
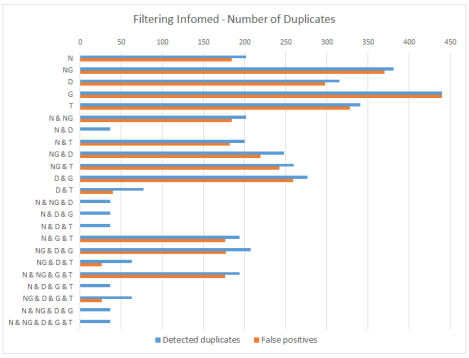


Figure 4: Rules for filtering Infarmed data source

We evaluated 30 aggregation rules which differ from one another in terms of complexity and variety. With those rules we realized that, for this data source, the property **Nome do Medicamento** was also the most relevant when grouped with others. Also, we realized that **Dosagem** is a good indicator when analyzing duplicates while **Genérico** does not help much. Furthermore and despite being a great indicator in this database, the **Dosagem** property works better when brought together with the property **Nome do Medicamento**. We also found that in this case, it is possible to find a perfect match between the number of duplicate results and the true quantity of duplicates in the database with only the conjunction of two properties.

Similar to Infarmed, the results have shown a proportional relation between the number of found duplicates and false positives and in terms of precision and recall we also concluded that as the number of duplicates increases, the precision decreases. Furthermore, it is possible to find high recall values even when false positives do exist - it is only need to find all the correct duplicates even though the filtering rule may detected more duplicates than the expected ones. Figure 5 shows some filtering rules examples for the dataset Infomed.



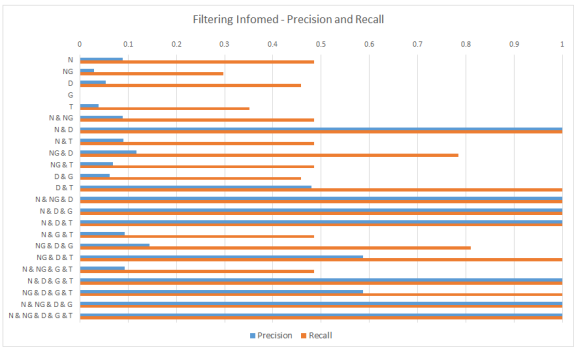


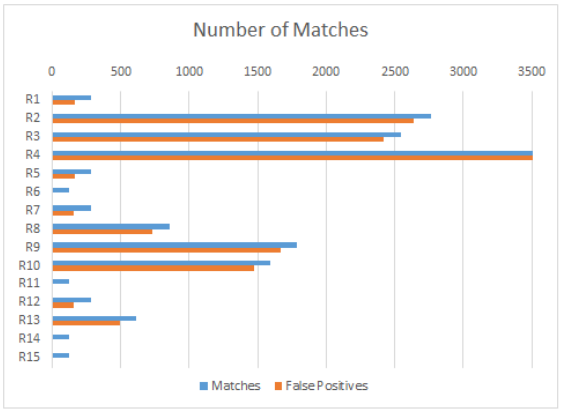
Figure 5: Rules for filtering Infomed data source

8.2 Mapping Data

Similarly to the filtering part of this work, in order to evaluate the mappings it was necessary to know how many matches there were between products from the two sources (Infarmed and Infomed). For that purpose, we analyzed which properties each source had and decided which pairs of properties could represent a match by looking to the values they assume.

Given the properties from Infarmed (Nome_do_Medicamento (N1); Substância_Activa (S); Dosagem (D1); Genérico (G1)) and Infomed (Nome_do_Medicamento (N2); Nome_Genérico (NG); Dosagem (D2); Genérico(G2)) we had assume the following 15 mapping rules, where the character "-" means "map":

- R1. N1-N2
- R2. S-NG
- R3. D1-D2
- R4. G1-G2
- R5. N1-N2 & S-NG
- R6. N1-N2 & D1-D2
- R7. N1-N2 & G1-G2
- R8. S-NG & D1-D2
- R9. S-NG & G1-G2
- R10. D1-D2 & G1-G2
- R11. N1-N2 & S-NG & D1-D2
- R12. N1-N2 & S-NG & G1-G2
- R13. S-NG & D1-D2 & G1-G2



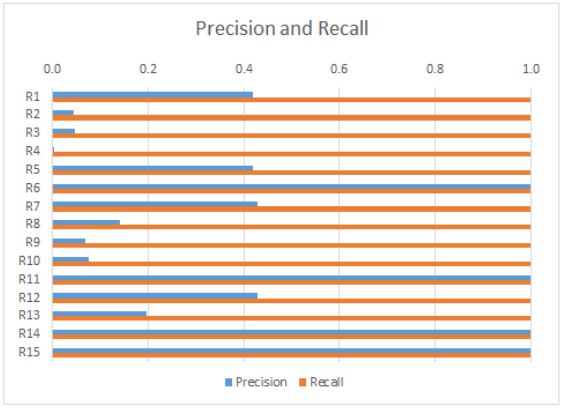


Figure 6: Rules for mapping Infarmed and Infomed data sources

- R14. N1-N2 & D1-D2 & G1-G2
- R15. N1-N2 & S-NG & D1-D2 & G1-G2

Each of these rules were implemented in the system with 252 products belonging to the first source and 405 belonging to the second source, 120 of which referring to the same real world object.

As shown in the Figure 6 the metrics used for this evaluation were the number of matches found, the number of false positives, precision and recall. Based on it, we can conclude that when mapping "Nome_do_Medicamento" with "Nome_do_Medicamento", each from one of the two sources, the obtained results are better than the ones achieved with mapping "Substância_Activa" with "Nome_Genérico". If we go further, we can consolidate that conclusion looking for the results differences between rules R6 and R8 - while R6 has a precision of 1, R8 presents a precision of only 0,14. In terms of precision, it is clear that having high number of matches does not mean that the mapping rule is better than the ones with lower number (comparing rules R4 and R15). The recall value shown to be always 1 because every rule found all the existing correct matches.

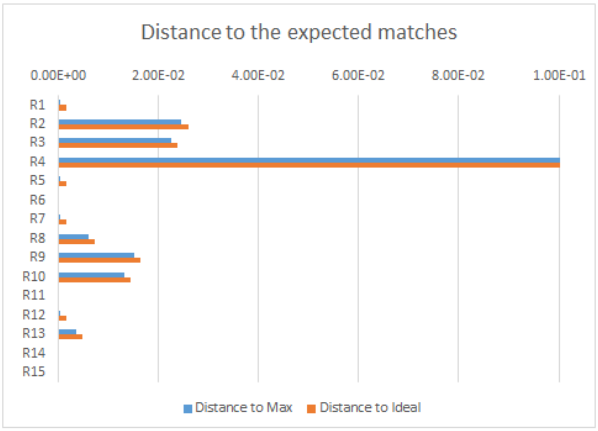


Figure 7: Distance to both, maximum and correct, number of expected matches

After this, we decided to introduce another evaluation metric called "Distance to expected match number" which tells us how many products could be matched at most: in our particular case, with 252 products from Infarmed and 405 from Infomed, we can only match 252 products at most. Therefore, every match that we get beyond the 252 are considered as excess. Besides that, and once we knew how many matches were truly expected, we have also calculated the distance to the real number of matches expected (shown in Figure 7).

This metric was conceived with the aim to help the curator to construct better mapping rules by letting her/him know the distance between the new mapping rule and the expected result, being that the lower the distance, the better the created rule.

According to the "Distance" metric, we can conclude that, for our example, the distances between the matches obtained in each rule and both the maximum possible matches and correct matches, do not differ a lot. Therefore, if we sort our rules by ascending distance value we will obtain the following list for both: [R6, R11, R14, R15, R7, R12, R1, R5, R13, R8, R10, R9, R3, R2, R4], where R6 represents the better rule while R4 is seen as the worse one to be applied to the system.

Regarding this study, the metrics "Number of matches returned" and "Precision" are inversely proportional while "Number of matches returned" and "Distance from the expected match number" are directly proportional. Thus, the main conclusion is that Curators should look for rules with number of matches close to the maximum expected number of matches, high precision values (close to 1) and distance values close to 0.

8.3 User Validation

In order to infer the application's utility and complexity, we decided to perform some tests with real

users. Those tests were made to twenty users, with a smaller version of the originally used database so that the users were able to understand the results and not be overwhelmed by tables with hundreds of lines. After doing the asked tasks in the system, the users answered to a brief questionnaire about the usefulness and complexity of the system.

8.3.1 Data Curator Users

In our work the Data Curator role is performed by a human that has the logical reasoning to decide which properties have to be identical to confirm that two RDF entities are referring the same real world product and can be merged. Therefore, the twenty users were asked to perform three different task with increasingly difficulty.

The first task was to find which property from the Infomed data source could pair with the **Substância_Activa** property from the data source Infarmed. This particular task was asked with the purpose of giving them a brief idea of how the mappings work and the users revealed a strong logical thinking, spending 10 seconds on average to find the right answer.

Before the users get to the point of mapping the data sources, they had to filter them. As it was explained before in this thesis, many times databases have duplicate information, for example, for the same smartphone model there are at least two different colors available (black and white) but if we think in terms of relevance probably the "color" feature is not relevant enough to determine that two products are actually different. Instead, there would be interesting to merged the information and present the user with only one entity that for the property "Color" assume both values combined "Black; White". That being said, for the second task, the users had to create a new filtering rule named "ByName" for the source Infomed, that aggregates products by the property **Nome_do_Medicamento**, choose that rule and verify the results. With this task the users had to understand what filtering by the property **Nome do Medicamento** meant and which results were expected. The users were able to recognize that the rule created by them was not a good one since it assume that two products are the same by only verifying if they have the same name.

In third task the users were asked to firstly filter each one of the data sources available, then to map those data sources by name and finally to verify the obtained results. With this exercise the users had to create a new mapping rule, with a specific syntax, that merged information of products from each one of the data sources that present the same value for the Name properties. Although this was the most time-consuming task, since the users had to do three different things (filter, create mapping and choose that mapping), they were able to verify that the results were the ones they were expected and also that once again, this rule was not as good as it seemed at first sight.

Almost all of the inquired said that the information given in the application was sufficient to understand and interact with the system. As it was expected the users revealed that the last task was the most complex one of the three while the first one was the easiest.

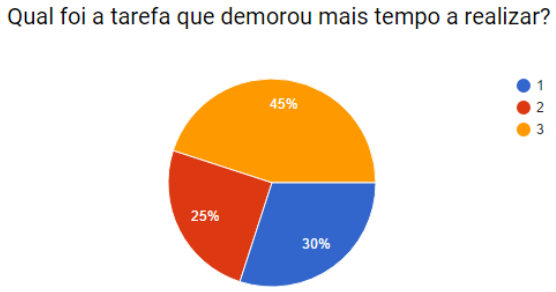


Figure 8: Curator users most time-consuming tasks

Overall, the users considered the application as being very useful for the daily life of a common user and that would like to have this system available in the contexts of: technological products, online stores and academic documents.

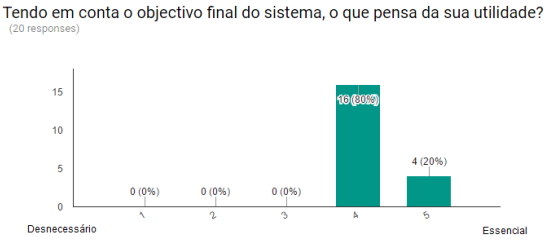


Figure 9: System utility in the Curators' perspective

With respect to the interface, the users suggested some small changes to improve the usability and understanding of the system, such as: simplify the way properties are presented and give more information whenever there are drop-down buttons.

8.3.2 Search Users

After the solution had satisfied the functional requirements of the architecture described in Section 7, it was given to twenty end-users to allow them to compare browsing for product information using the traditional web search and our semantic-aware approach. Since these type of users could be common people, without background on databases or computer science, the tasks chosen for them were

relatively easy. They only had to simulate a choice of an existing mapping rule that better suited their needs, and then they had to perform a search, via the application, by attributing a value to a specific property.

For the first task, users had to make two consecutive searches with the same value for equivalent properties but in each of the different data sources. This task has served to show to the users that if they search for Betadine in Infomed they will get too few information, while if they search for the same product in Infarmed they will get much more information. Besides that, the product returned was common in both databases - that was made with the purpose of asking them if it would make sense to merge the two products and offer a more complete and unified information.

In the second task, users had to map the two sources by a rule previously created by us, then they had to perform the search performed in the previous task but in the "All" model and had to verify the results. With this task the users were able to see the problem found in the first task solved, i.e. in the end they could see two products Betadine, one from Infomed that mapped with another from Infarmed and also the other Betadines that still exist but that did not map with other products. With this task the users revealed a further understanding about the problem stated in this thesis and a confirmation of the returning results.

Finally the users were asked to test the application to infer if it was capable of searching by more than one property at a time. That way, the users were asked to choose a previously created mapping rule to be applied to the system, and then to perform a search by two properties and verify the results. With this task the users got to know the system better in terms of how the mapping is done and see different features of it.

In average, the search users spend more time in the first task which means that once they are comfortable with the system they could be quicker and that the system is easy to learn and to interact with.

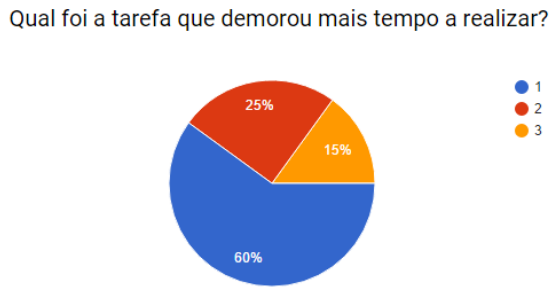


Figure 10: Search users most time-consuming tasks

According to the questionnaires, most of the users said that they did not need more information to understand which fields they have to select to perform the tasks. Also, more than half of the inquired stats that all the tasks were simple and almost all of the inquired answered that the system had great usefulness in their lives.

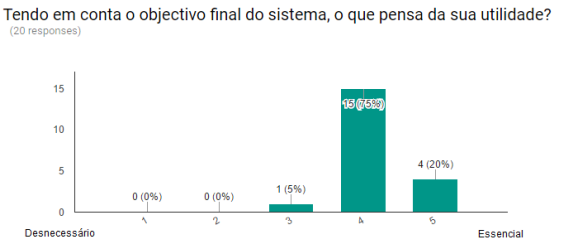


Figure 11: System utility in the Search users' perspective

9 Conclusions and Future Work

The Linked Product Data solution came to address the identity mapping and user's workload problems since it provides tools to structure data about products. Therefore, the aim of this work was to facilitate the user's searches in a way that s/he will not has to be concerned about which sources describe the product that s/he was looking for. Instead, all the information needed was presented in a single place rather than in a considerable number of distinct web pages.

The work presented in this document provides as a major achievement a way to address the problems of identity and redundant information since similar data is kept in distinct databases, which have different IDs for the same real products. In order to overcome these obstacles, we first suggested an examination of the available data sources by an expert user called Data Curator, with the knowledge to create rules that will aggregate duplicated data and map information from disparate data sources.

After solving these obstacles we were able to offer common users an interface, on one specific context (pharmaceutical products or cinematographic works) at a time, where they can search in a property-value perspective and obtain the results in table format where all the information appears unified - the one that was merged and the remaining information from each one of the data sources.

According to the users evaluation, we were able to verify that the users workload was reduced, that the designed interface is not too complex to be understood, although there are still some improvements to be taken into account, and that it is seen as a tool of great usefulness for people who perform searches daily.

9.1 Future Work

Although there were interesting results, there is still some work to be explored in this context. For example, the last contribution can be used to evaluate mapping rules created automatically by the computer. A semi-automated approach could be developed, where the Data Curator is responsible for the evaluation of the rules created automatically by the program, which means that he only will have to decide which rules best suits his necessities.

For an automated approach, the computer computation would be the responsible for creating the mapping rules based on all possible combinations of existing properties and then, those rules can be evaluated by the created metric to infer accuracy.

Finally, it would be interesting to add the Good Relations Ontology [8] or other ontologies, already used for e-commerce, to structure the data from the beginning of the whole process using existing classifications.

References

[1] B. Alexe, L. Chiticariu, R. J. Miller, and W. C. Tan. Muse: Mapping understanding and design by example. In *Proceedings - International Conference on Data Engineering*, pages 10–19, 2008.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a Web of open data. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4825 LNCS, pages 722–735, 2007.

[3] K. Belhajjame, N. W. Paton, S. M. Embury, A. A. Fernandes, and C. Hedeler. Feedback-based annotation, selection and refinement of schema mappings for dataspaces. *EDBT '10 Proceedings of the 13th International Conference on Extending Database Technology*, pages 573–584, 2010.

[4] S. Brin and L. Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 56:3825–3833, 2012.

[5] H. Cao, Y. Qi, K. S. Candan, and M. L. Sapino. Feedback-driven result ranking and query refinement for exploring semi-structured data collections. In *Proceedings of the 13th International Conference on Extending Database Technology - EDBT '10*, page 3, 2010.

[6] G. Friedrich and K. Shchekotykhin. NameIt: Extraction of product names. *Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06)*, pages 29–33, 2006.

[7] O. Hassanzadeh and M. Consens. Linked movie data base. In *CEUR Workshop Proceedings*, volume 538, 2009.

[8] M. Hepp. GoodRelations: An ontology for describing products and services offers on the web. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5268 LNAI, pages 329–346, 2008.

[9] R. Isele and C. Bizer. Learning expressive linkage rules using genetic programming. *Proceedings of the VLDB Endowment*, 5(11):1638–1649, July 2012.

[10] H. Köpcke, A. Thor, S. Thomas, and E. Rahm. Tailoring entity resolution for matching product offers. *Proceedings of the 15th International Conference on Extending Database Technology - EDBT '12*, page 545, 2012.

[11] R. McCann, W. Shen, and A. Doan. Matching schemas in online communities: A Web 2.0 approach. In *Proceedings - International Conference on Data Engineering*, pages 110–119, 2008.

[12] H. Pinto, A. Gómez-Pérez, and J. Martins. Some issues on ontology integration. *Proceedings of IJCAI99's Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends*, 1999:1–12, 1999.

[13] Y. Raimond, C. Sutton, and M. Sandler. Automatic interlinking of music datasets on the Semantic Web. *CEUR Workshop Proceedings*, 369, 2008.

[14] R. Ramezani. *SWApriori : A New Approach to Mining Association Rules from Semantic Web Data*. PhD thesis, Isfahan University of Technology, 2012.

[15] R. Ramezani, M. Saraee, and M. A. Nematbakhsh. Finding association rules in linked data, a centralization approach. *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, pages 1–6, May 2013.

[16] D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, and M. Hassanali. DrugBank: A knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research*, 36, 2008.

[17] L. L. Yan, R. J. Miller, L. M. Haas, and R. Fagin. Data-driven understanding and refinement of schema mappings. *ACM SIGMOD Record*, 30:485–496, 2001.