A Semantic Method for Multiple Resources Exploitation

Abdullah Almuhaimeed
University of Essex
Wivenhoe Park, CO4 3SQ
Colchester, United Kingdom
ansalm@essex.ac.uk

Maria Fasli
University of Essex
Wivenhoe Park, CO4 3SQ
Colchester, United Kingdom
mfasli@essex.ac.uk

ABSTRACT

Being able to extract and exploit information that is included in multiple resources (repositories, corpora, etc.) is essential to benefiting from the increasing availability and complementary nature of such data scattered across the World Wide Web. However, such an endeavour raises a number of challenges including dealing with the diverse structures of such resources, different relationships among such data, and the overlapping and complementary nature of the information. Thus, developing a semantic method that can extract semantic information and hidden associations would help overcome such difficulties that occur when dealing with multiple resources. This paper presents a new semantic method that exploits the overlap between various resources with different structures (i.e. ontologies as forms of structured data and corpora as examples of unstructured data) and employs semantic relations, specifically sibling relations, to infer new information that may not exist in the original resources. Then, this method employs the new information in a content-based recommender system to enhance the quality of the provided recommendations (i.e. articles) in complex fields that are inherently characterised by varying relations and structures, such as bioinformatics. In addition, this method is accompanied by an automatic tool that is responsible for tailoring individual recommendations to each user based on his/her profile.

Categories and Subject Descriptors

H.4 [**Semantic-Techniques**]: Recommendations and Reasoning

Keywords

Semantic Web, Recommendations, Personalisation and Bioinformatics

1. INTRODUCTION

The abundance of resources on the World Wide Web with varying structure which may contain potentially complimentary information has led to an increased need to develop semantic techniques that can handle these resources, extract semantic relations and associations, then employ the extracted information to enhance services, such as searching for information and recommendations. However, extracting semantic relations and associations that occur as a result of overlapping or complementary information is a non-trivial task because these resources have varying structures, e.g. ontologies, or have no formal structure as in corpora. This represents a challenge that should be addressed as part of developing semantic-based techniques. There are relevant studies that have tried to solve similar problems by exploiting semantics included in multiple resources that are represented by a Linked Open Data (LOD)[1] dataset, which is an ontology that has been formulated from multiple resources. However, these works still suffer from some shortcomings in exploiting the semantics between resources. For instance, Mirizzi et al. [15] provided a recommender system for movies based on a LOD dataset in which they tried to exploit semantics between different concepts included in the ontology to provide recommendations on movies. This work was further enhanced by [4]. The authors made a slight change in the method used to recommend movies by changing the method of calculating the similarities between movies to enhance the recommendation accuracy. Even though both works have used semantics acquired from this ontology, they did not exploit triples to infer and extract new semantic information that can be used to enhance the quality of the recommended items. Also, neither of the approaches exploited the ontology in constructing user profiles and in addition their profiles were not adaptable because they lacked updating and deleting methods. Hence, the aforementioned issues lead to inaccuracy in the recommendations provided through these approaches. Another related work is presented in [13] that involves a recommender system that provides recommendations for online academic papers. It uses a single source (i.e., ontology) to enrich a user profile and draw recommendations based on this enrichment. Moreover, this approach considers user feedback to construct the user profile. Two recommender systems are constructed in this approach: Quickstep and Foxtrot. Quickstep exploits ontological deduction to enhance the profile and uses an external ontology that improves user profiling. It considers a research paper topic based on the ontology of computer science classifications performed by a directory from the Open Directory Project (ODP).[2] This approach used a k-Nearest Neighbour

SEMANTiCS '15, September 15-17, 2015, Vienna, Austria

DOI: http://dx.doi.org/10.1145/2814864.2814868

[1] http://linkeddata.org/

[2] http://www.dmoz.org/

classifier for semantic annotations in the paper, associating them with the topic of the paper, including the ontology. The latter (Foxtrot) enhanced Quickstep by employing user feedback. This system does not take into account the availability of multiple resources of information (ontologies, taxonomies, etc.) to enrich the user profile, which may decrease the accuracy of the provided recommendations.

This paper introduces a new method that can reason through various bioinformatics resources and extract semantic relations, such as siblings and associations, to infer new information that can be utilised to enhance the quality of the recommended items (i.e. articles). This service can be distinguished by its ability to exploit some semantic relations, such as sibling, which have been gained as a result of information overlapping between different concepts from multiple resources and thus enhance the accuracy of the provided recommendations. Moreover, this method has been supported by a fully automated adaptive ontological user profile tool that can tailor individual recommendations based on user preferences and, therefore, support users with the most relevant articles that align with their preferences. The aforementioned methods showed some improvement on the level of accuracy for the recommended items; this improvement has been assessed based on user-centric evaluation experiment that has been applied to a set of participants who specialise in bioinformatics related areas.

2. BACKGROUND

In this section, we provide a brief description of related work on ontologies, reasoning, recommendations and personalisation.

Ding et al. [6] argued that ontologies are fundamental to the Semantic Web as they encode information about the underlyling domain. Ontologies can be constructed using several languages, including the Resource Description Framework (RDF), Resource Description Framework Schema (RDFs) [14], and the Web Ontology Language (OWL) [11]. Such languages need to have specific features; namely, (i) conceptualisation, wherein language should follow a specific model to represent facts included in the ontologies; (ii) vocabulary which describes syntax and grammar; (iii) axiomatisation, which is used to describe the language's rules and constraints [6]. Reasoning mechanisms can then be employed to reason over the information encoded within ontologies.

Tari et al. [18] introduced a method that discovers new drugs through automated reasoning. This method exploits knowledge and facts to discover new drugs gained from literature and knowledge bases. This reasoning is performed in *AnsProlog* by scrambling molecular effects that are the results of drug-target interactions linking diseases and drug mechanisms as domain knowledge. This method is based on three factors to perform automated reasoning and discover new drugs: knowledge acquisition, which involves selecting a source and extracting facts that help identify drug identifications gained from text mining; knowledge representation, which acquires logic facts from various resources and logic rules that represent a drug mechanism's properties; and knowledge reasoning, which links several sources and allocates order to the steps that lead to drug indications. Thus, this approach performs reasoning based on the action description and logic of fact.

As very often in searching for information or providing recommendations these need to be tailor-made to the user, user profiles represent an important source of information that can help provide users with personalised recommendations based on their individual preferences. Information stored in user profiles can be classified into two types: static and dynamic. The former contains information that does not change frequently; for example, name, age, job, etc. The latter represents information that changes frequently such as preferences. Mezghani et al. [12] describe how user preferences can be collected in two ways: explicitly, in which users are asked to answer questions and fill forms, or implicitly, where a tool installed on the user machine is able to capture user behaviour implicitly without any burden to the user. Each of the aforementioned methods has its pros and cons. For instance, explicit collection is easier and quicker for collecting user data, but it is inaccurate and causes some inconvenience to users by requiring them to answer questionnaires and fill in forms. However, implicit collection is difficult to execute, since it takes time for this method to gather enough data to provide accurate information about user preferences. Nevertheless, it draws more accurate conclusions about each user's preferences to provide him/her with accurate recommendations services [19].

Yoneya and Mamitsuka [20] introduced a new recommender system, PURE, which is based on content-based filtering. Their approach is focussed on providing recommendations for PubMed[3] articles. PURE has an interface that allows users to interact with it and add/delete their preferred articles daily. Thus, each user needs to create a profile to be served by PURE. This system clusters the articles provided by the users, then sends daily e-mails about the new articles to them in the form of recommendations.

3. A SEMANTIC REASONING METHOD

Our work has been motivated by the need to help specialist users, such as researchers, to find content that is relevant to what they wish to read and due to the abundance of available resources search may be complicated, especially in complex fields like bioinformatics. This also motivated us to develop techniques that are able to overcome the challenges that overlapping information in resources entail and indeed exploit this overlap to provide users with the most relevant content. Thus, we developed a semantic-based method that extracts semantic relations and hidden associations from bioinformatics resources (e.g. ontologies such as Protein Ontology (PO),[4] Gene Ontology (GO),[5] Open Directory Project (ODP)[2], and Bioinformatics Links Directory (BLD),[6] as well as corpora (such as Wikipedia[7]) that concern the target audience for this work. Since the aforementioned resources include a variety of information that could be exploited to support users with better recommendations that fit into their preferences, this is the main goal of this research. Moreover, this work has been extended with a method that collects user preferences implicitly, then calculates a similarity score between them and uses the ODP ontology to construct an ontological user profile for each user, which contributes to enhancing the quality of the provided recommendations. These methods will be illustrated by supporting bioinformaticians' work with recommendations of

[3] http://www.ncbi.nlm.nih.gov/pubmed
[4] http://pir.georgetown.edu/pro/
[5] http://geneontology.org/
[6] http://bioinformatics.ca/links_directory/
[7] http://www.wikipedia.org/

the most relevant content (i.e. articles) from BMC,[8] which is a bioinformatics corpus.

To achieve these goals, we need to design a reasoning method that is able to exploit the overlapping and complementary information between various resources in the field of bioinformatics, such as PO, GO, etc., and then discover semantic relations such as siblings and hidden associations between concepts. Then, we will utilise SPARQL[9] queries to mine information and this represents a reconciliation or alignment step between different resources, because it selects similar type of content from each resource such as class, subClassOf,etc. Next, our developed method will hand the extracted data to the reasoner[10] which employes the semantic rules in tables 1 and 2 and the reasoner will infer all semantic relations (i.e. sibling, this is one of the main relations that can be discovered through reasoning, but others are considered too, to identify the ones that provide the best improvement to the recommendations). Subsequently, a semantic network will be drawn that represents the inferred relations and information. Our assumption for constructing the semantic network was that every orphan concept is a direct subClassOf *Thing* and it will be listed under superClassOf *Thing* as well. This process will perform a direct mapping between concepts to remove duplicate concepts from the semantic network. A set of contents reasoned for each class will then be added to the processed class or concept, such as lists of subClassOf, superClassOf, and siblings. After that, each class becomes a subject in one or more triples. Thus, to represent our semantic network with all inferred relations and information, the Dijkstra [5] algorithm was modified to fit with our resources. This algorithm was designed to calculate the shortest path between two points in a graph. We altered this method to calculate the shortest path from the class "*Thing*" instead of calculating the path from any point in the graph. The reason for this change to Dijkstra's algorithm was to assign weight to edges (i.e. links) that connect different concepts in our semantic network; the link weights started at 1, and the number decreased by 0.25[12] whenever the concept took a step away from class *Thing*. This calculation will help us to know how far a given concept is from the class *Thing*.

Further on, the user profiles will be enriched with the most relevant information that can be gained from the semantic network; this enrichment will be exploited to enrich each user's query by supporting him/her with valuable information gained from the processed resources. Our hypothesis is that the aforementioned enrichment contributes to improving the level of accuracy of the provided recommendations and results. The contribution of this work comes through its ability to extract semantic relations, such as siblings and hidden associations, from multiple resources that have different relations and information, and method exploits the information overlap to enhance the accuracy of the provided recommendations to each user based on his/her profile. This presents different challenges that need to be overcome, such as dealing with different structures and variety in relations which are not easy tasks. Our recommendation method in essence contributes to content-based techniques. Finally, we construct an adaptive ontological user profile based on the ODP ontology (branch of informatics) in order to provide better recommendations to each user. This profile should be supported with methods that are able to add, delete and update users' preferences automatically, without any intervention from the user, and it should be able to tailor recommendations to each user based on their preferences.

Table 1: Semantic Rule Terminologies

Semantic Rule Terminologies
Classes := C_1,...., C_n
C :=name, subClassOf, Comment, label,
equivalentClassOf,objectProperty
subClassOf := C ∈ Classes
Restrictions := onPropert,objectProperty,
SomeValuesFrom
onProperty := C ∈ Classes
SomeValuesFrom := C ∈ Classes

Table 2: Sibling Rule

Sibling Rule
Requirement: *List of InfModel*
Input: *Selected Data (Classes and Classes' Comments)*
Output: *List of concepts*
Rule : This rule shows that x is sibling of y:
sibling(x,y) $\implies$ x ∈ Classes ∧ y ∈ Classes ∧
∃ z (z ∈ Classes) ∧ z ∈ x.subClassOf ∧ z ∈ y.subClassOf

3.1 Recommender Services Construction

The enhancement of recommendation services is the primary intended goal of our developed method. This enhancement will be achieved by creating an inference method between different resources, which is able to extract semantic relations and hidden associations between these resources, and then represents the gained relations and information as a semantic network. Then, it will connect semantic network concepts with the most similar content in each user profile in order to enhance the accuracy of the provided recommendations. To achieve this target, a prototype system has been developed that helps bioinformaticians find articles in the BMC corpus. The Lucene Search Engine[13] was used for indexing and retrieving articles from the BMC corpus. A set of ontologies and Wikipedia have been employed to extract semantic information for users' query enrichment. A method has been developed that is able to collect user preferences automatically and implicitly, then it calculates similarity between the ODP ontology and the user's preferences to construct an ontological user profile. A set of re-ranked articles will be returned and organised based on their similarity to both the user's preferences and semantic enrichment. This system gives users the opportunity to narrow down recommendations by selecting a specific interest to receive recommendations exclusive to the elected preference. Algorithm 1 provides an outline of the high level process performed by the prototype system.

[8] http://code.google.com/p/bmc-bioinformatics-processed-corpus/

[9] http://www.w3.org/TR/rdf-sparql-query/query

[10] We have used a standard reasoner included in Jena[11] framework to infer new relations and associations through different resources.

[11] http://jena.apache.org/documentation/inference/

[12] We have considered this number as suggested in[17], the reason behind selecting this particular value is that the average of neighbours for each node in our semantic network equals 4, and this means one divided by four.

[13] http://lucene.apache.org/core/

Algorithm 1: Content-Based Recommendation Services

```
Data: User's Query and User's ID
Result: List of Recommended Items
Submit(query);//To receive query.
Enriched_query =
Enriched_with_required_relation(query,User_ID); //This to
enrich user's query with semantic relation concept.
lucene_Search_engine(Enriched_query,100); //This will send
user's enriched query to get number of hits results.
//The following loop will go through all articles.
for returned_Article do
    //The following loop is to read files.
    while counter <returned_Article.length do
        Array_of_Strings = read(returned_Article);//Read file
        and store it in array.
    end
    //The following line will Convert file into vector to
    calculate similarity.
    V2 = Convert_query_result_to_vector(Array_of_Strings);
    Get_User_preferred_Concept(User_ID); //Retrieve all user's
    concepts.
    //The following loop goes through user's concepts.
    for each User's Concept do
        SemanticEnrichedConcept= (userConcept.Description +
        SN_enrichment);
        //The following line will convert user's concepts
        description and semantic enrichment into vector to
        calculate similarity.
        V1 = Con-
        vert_User_concept_to_vector(SemanticEnrichedConcept);
        sim = V1.getCosineSimilarityWith(V2); //Calculate
        cosine similarity between two vectors.
        Document_Simi_Score +=termWeight() * sim; //Total
        similarity for each file.
    end
    Document_final_Score = Document_Simi_Score *
    lucene_Score;
    hitsMap.put(queryFilePaths,Document_final_Score); //Fill
    Hashmap with file paths and score of similarity.
end
//The following loop will add and update and show
recommendations on top 30 preferred articles.
for each Result in hashMap do
    Index = 29;
    add_to_userprofile(Result);//Save recommendaed articles in
    the user profile.
    Index −;
    if Result in 30th preferred Articles then
        Update_Userprofile_score(Result);//Update document
        score.
        Show(Recommended_Articles); //This shows
        recommended articles based on similarity with user
        preferences.
    else
        add_to_userprofile(Result);//Save recommendaed
        articles in the user profile.
        Show(Recommended_Articles); //This shows
        recommended articles based on similarity with user
        preferences.
    end
end
```

3.2 User profile

Constructing ontological user profiles requires several data pre-processes in order to offer each user individual recommendations based on his/her preferences or interests. A typical problem faced by content-based systems is the cold start problem; this is when there are not enough user data. Determining the needed information is an important task to start constructing our user profile. Our approach collects user data from three different entries; namely, surfed URLs, clicks and bookmarked Web pages. So, from the aforementioned entries, the preferences can be formulated by assigning a weight, called Term Frequency, to each concept; the concept's weight reflects the importance of the concept to the user. Moreover, to make the created user profile adaptable, there is a mechanism that allows our user profile to be updated for each user, since it has been equipped with add, delete, and update methods that can be run automatically. This mechanism will be discussed in detail in the following sections.

3.2.1 User profile Management

The component responsible for adding the interests to the user profile performs four main steps. Firstly, the *ManicTime*[14] tool works automatically when the browsing process starts and collects the URLs, time spent in each URL, and the date that URL was accessed. A plug-in that is connected to the browser (i.e. works side by side with ManicTime) collects interactions that ManicTime is unable to collect, such as bookmarks and number of clicks on each website. Secondly, all these data are collated and the interests are added by applying the following equation:

$$User's\,Term\,Frequency = \frac{Simi_i+Frq_i+Nv_i+T_i+B_i+C_i}{Simi_i+Frq_i+Nv_i+T_i+B_i+C_i+a}$$

$Simi$ represents the cosine similarity score that the surfed URL satisfies with the ontology concept. Frq represents the frequency of URL that has some similarity to the ontology concept. Nv represents the number of visits to the URL. T represents the total time spent reading the concept. B represents whether the webpage is bookmarked or not; this takes a value of 1 when the page has been bookmarked by the user and 0 if not. C represents the number of clicks in this website; a is a constant equals 100, where this number is the best value we have reached to make the $User's\,Term\,Frequency$ between 0 and 1. The results are normalised by dividing each result by the numerator value plus a hundred. Finally, i represents the preferences stored in the user profile. In the third step, after calculating the term frequency for each term, all items with frequency values above 0.1 (i.e. the thresholds were identified based on several runs, and this threshold provided the best recommendations) are stored with recent visits, which reflect the last time that the user visited a specific URL; this information can be important in specific situations. For instance, let us assume that a user has the same term frequency score for two different concepts, but the times of recent visits for the two concepts are different. So, in such a case, the times of the recent visit will reflect the importance of each concept in comparison with the other, to determine which concept is preferred by the user. In the fourth step, update and delete mechanisms are applied to generate updated user profiles that better represent the user preferences. So, the update mechanism will increase term frequency (term weight) for each visit by 0.05 (i.e. the thresholds were identified based on several runs, since this amount increases to fit with the number of maximum days since the last visit, which is 20 days), which represents the amount of daily increase and decrease. Therefore, when the website is visited by the user, it will increase and the opposite will take place otherwise. Moreover, the delete mechanism will run when the user's visit length is less than a threshold of 10 seconds or the number days since the last visit is more than 20 or term frequency is less than 0.1. The deleting or forgetting mechanism will be discussed in more detail in the next section, where all reasons behind considering these thresholds for

[14] http://www.manictime.com/

deleting preferences will be clarified. Algorithm 2 provides more detail about managing the user profile (add, update and delete).

3.2.2 Forgetting Mechanism

Some users show no interest in some concepts and therefore, such concepts should be removed from the user's profile in order to provide more accurate results that do not contain old interests. There are some factors that indicate a specific concept becomes unwanted and needs to be deleted. As the work of [10] suggested, since the minimum time spent by a user on a webpage is between 10-20 seconds, and since this period reflects whether the user is interested in the webpage or not, therefore, when a user spends more than the suggested time on a webpage, this means he/she is interested; and, otherwise, it means he/she is not interested. As a result, we established 10 seconds as our threshold and hence, when a user spents less time on a webpage than the threshold, it meant that he/she was not interested in the webpage. Furthermore, for the URLs that have not been visited by the user, we suggested an initial forgetting mechanism that will use the current days since last visit norm in browsers; i.e. 20 days. Thus, if a URL has not been visited by the user for 20 days, then the interest will be deleted. The second factor, which is term frequency, involves decreasing the term value in time, in case the user does not re-visit the URL, until it reaches our threshold (0.1), and then it will be deleted by the deletion component. Algorithm 3 will be run after the previous algorithm (2) just to make sure that all preferences' weights and dates fit with our determined thresholds.

Algorithm 2: Managing Concepts in the User Profile

```
Data: User's ID, URL_OPD_Similarity, Term_Frequency,
      Num_of_Visit, Total_time_Spent, Num_of_clicks,
      Bookmark and Date_of_lastVisit
Result: List of preferences
//Adding Preferences to the user profile
for counter <number_of_matched_URL_and_ODP do
    User's Term Frequency = (Simi_i+Frq_i+Nv_i+T_i+B_i+C_i) / (Simi_i+Frq_i+Nv_i+T_i+B_i+C_i+a)
    add_preference(); //This step is to add user's preference
    and match it with most similar concept to exploit sibling
    relation
    counter++;
end
//Managing and updating preferences in the user profile
date_of_data_collection = Recent_date_in_user_profile;
theSmallest_weight_userPrifile_should_have = 0.1;
number_of_days_since_last_visit = 20;
Daily_decrease_weight = 0.05;
Daily_increase_weight = 0.05;
TheLowestDuaration = 10;
for each Preference in User_Profile do
    //The following line will retrieve set of details about url
    such as date of visit, total time in all visit and user ID
    UserProfile = details about visited website;
    Update_wieght_for_url(Preference, UserProfile,
    date_of_data_collection,
    theSmallest_weight_userPrifile_should_have,
    Daily_decrease_weight, Daily_increase_weight,
    number_of_days_since_last_visit)
    currentWeight = Specific_user_urls.getTerm_frq();
    formatter = new SimpleDateFormat("yyyy-MM-dd");
    diffTime = 0;
    DifferenceIndate = 0;
    tempWeight = 0.0;
    // This loop is to update weights and delete unwanted
    preferences
    for i <UserProfile.size() do
        log1 = UserProfile.get(i);
        dateInString = log1.getDayOfVisit();
        date1 = formatter.parse(dateInString);
        if (i + 1) ≥ UserProfile.size() then
            date2 = formatter.parse(date_of_data_collection);
            diffTime = date2.getTime() - date1.getTime();
            DifferenceIndate = diffTime / (1000 * 60 * 60 *
            24);
            tempWeight = (currentWeight - (decreaseValue *
            DifferenceIndate));
            if (DifferenceIndate ≥ maxNumofDays) or
            (tempWeight ≤ smallestValue) then
                currentWeight = tempWeight;
                Delete(Preference);
            else
                currentWeight = tempWeight +
                increaseValue;
                Update(Preference);
            end
        else
            log2 = UserProfile.get(i + 1);
            date2 = formatter.parse(log2.getDayOfVisit());
            diffTime = date2.getTime() - date1.getTime();
            DifferenceIndate = diffTime / (1000 * 60 * 60 *
            24);
            tempWeight = (currentWeight - (decreaseValue *
            DifferenceIndate));
            if (DifferenceIndate ≥ maxNumofDays) or
            (tempWeight ≤ smallestValue) then
                currentWeight =
                Specific_user_urls.getTerm_frq();
            else
                // This line to increase the weight.
                currentWeight = tempWeight +
                increaseValue;
            end
        end
    end
end
```

4. SEMANTIC METHOD AND RECOMMENDATION SERVICES EVALUATION

We conduced a user-centric evaluation that involved 30 human participants who were experts in bioinformatics. Each participant was assigned to a group and interacted with a system that was being evaluated, following the method of Knijnenburg et al. [9], which was used to evaluate the level of enhancement achieved when considering the semantic relation (i.e. sibling) in our recommender approach. Their method follows five steps for assessing the recommender system. (i) Randomly select the set of participants, explain the purpose of the experiment and divide them into groups that contain an equal number of participants. (ii) Each evaluated system should have a single group. Then, (iii) participants should be asked to interact with five well-defined tasks as suggested in [3] and save the user's interaction in the database. Then, (iv) ask the participants to fill out a questionnaire (3) to determine their opinions regarding the different functionalities provided in the evaluated system. Finally, (v) analyse the participants' data to measure the performance of each evaluated system.

The five systems evaluated are: (i) our approach with all features (sibling relation as well as adaptive ontological user profile), called BioRec_Full; (ii) our approach without the created semantic network, called system BioRec_SN; (iii) our approach without the user profile (with sibling relation only), called system BioRec_Profile; (iv) a method from the literature [15] and more specifically the Mirizzi system (this was designed to provide recommendations on movies and extract semantic information from LOD[1], then use them to enhance the quality of the provided recommendation. The Vector Space Model (VSM) [16] is used to handle semantic information that exists between LOD[1] concepts. Furthermore, this approach considered other factors such as genre

Algorithm 3: Forgetting Concept from the User Profile

```
Data: User's Preferences
Result: Set of preferences without unwanted items
//This to delete any item with weight less than threshold or
number of days since last visit is greater than 20
for each Preference in User_Profile do
    if (Preference.getWeight <0.1 ) or
    (Preference.getLastVisit ≥ 20) then
        Delete(Preference);
    end
end
```

(i.e. comedy) and weight for each property, represented by α p and assigned by weight to represent the level of importance for a feature to the user. Since these factors do not fit with our approach as our articles do not have these properties, we have used a modified version in which these features were removed); (v) the baseline approach which is Google API[15,16] indicated by as system Google. The data was collected and stored in four stages: (i) met with participant and installed plug-in (i.e. collect user interactions automatically) and ManicTime on his/her machine; (ii) collected data from the participant's machine and created dynamic ontological user profile; (iii) collected data that represented user interactions (clicks, rates, etc.) with the systems while performing the five assigned tasks; (iv) collected the questionnaires.

Our evaluation method concentrated on the classification-accuracy metrics. Participants were asked to choose four values to rate each result from the top 30 results (because it was difficult for users to rate all of the provided results), where "highly relevant" equals 4, "relevant" equals 3, "relevant to some extent" equals 2, and "not relevant at all" equals 1. These ratings have been applied on all participants from all comparative approaches. We only considered score 4 "highly relevant" as good recommendation and considered the other scores as bad recommendations. Then, the mean average precision [1], was considered in order to assess the level of success achieved in all of the comparative approaches, this metric and precision at N are useful for recommender systems [8], especially for recommender systems with pre-ordained nature [2].

Furthermore, our questionnaire that covered the five compared approaches consisted of eight statements that help to evaluate the level of success achieved by a recommendation system. The questionnaire statements were: (1) the items recommended to me matched my interests; (2) the recommendations provided to me were useful; (3) overall, I have been satisfied with the services provided by the recommender system; (4) the items recommended to me are diverse; (5) the recommendation I have received better fits my interests than what I may receive from other search or recommender systems that I have used in the past; (6) the recommended items cover a broad range of specific interests that I have in this area; (7) the recommender system helps me to discover new articles; (8) if a recommender such as this exists, I will use it to find articles to read. Half of the statements (1, 2, 7 and 8) were taken from [7] and [9], as these statements are general to some extent and should exist in most of the recommender systems to measure the different factors (e.g., diversity, accuracy, novelty and satisfaction). The remaining statements (i.e. 3, 4, 5 and 6) were tailored to assess specific features in our recommender system. Thus, these statements have a scale from 1, "Strongly Disagree" to 5, "Strongly Agree". Therefore, through these eight statements and by calculating the mean average precision (MAP) for each participant through all of the comparative approaches, we can assess the level of success in enhancing the preciseness of recommendations in our recommender approach in comparison with the other approaches.

[15] https://cloud.google.com/prediction/docs

[16] https://developers.google.com/custom-search/?hl=en

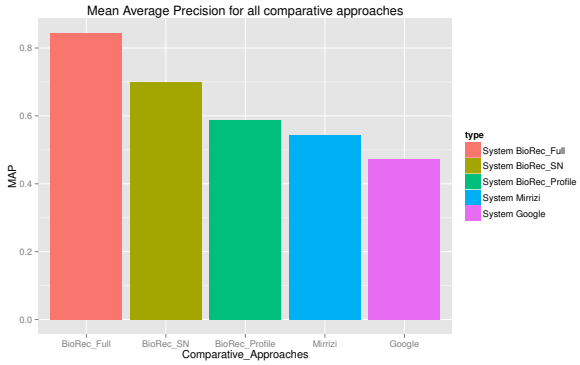

Figure 1: MAP Metric for Bioinformatics Recommender Services.

Thus, as shown in table (3), participants in group BioRec_Full (system BioRec_Full's participants) were satisfied with most of the functionalities provided by system BioRec_Full. System BioRec_SN's participants' were the second most satisfied. System BioRec_Profile's and system Mirrizi's results were close or similar to each other in some statements, such as statement 2, statement 5 and statement 6. Finally, system Google registered the lowest score, and this reflects the fact that participants in this system only receive generic recommendations.

5. RESULTS

This experiment has helped us assess various aspects of the developed methods when applied to multiple bioinformatics resources.

Figure 1 shows that system BioRec_Full outperformed the other comparative approaches by a significant score. This reflects the importance of an adaptive ontological user profile as well as semantic network enrichments with sibling relation over multiple bioinformatics resources. System BioRec_SN, which represents the second system in comparison with the remaining approaches (BioRec_Profile, Mirrizi and Google), shows how a dynamic ontological profile enhances the precision of the recommendations provided. Moreover, systems BioRec_Profile and Mirrizi registered close results, but system BioRec_Profile outperformed both, thereby reflecting the importance of semantic network enrichment and its ability to enhance the preciseness of the recommendations provided. Finally, system Google had the lowest results because it provides standard recommendations rather than personalised ones. The t-test metric was applied to the results gained from the MAP metric, and it registered significant results. For instance, the t-test between system BioRec_Full and system BioRec_SN registered a score 0.010, which was considered a significant difference on the t-test scale, the score between system BioRec_Full and system BioRec_Profile registered a score 0.00000726, the score for between BioRec_Full and Mirrizi was 0.0000162, and finally the score between system BioRec_Full and system Google

Table 3: Questionnaire Evaluation for Bioinformatics Recommender Service (Stm: Statement)

Systems	Stm 1	Stm 2	Stm 3	Stm 4	Stm 5	Stm 6	Stm 7	Stm 8
BioRec_Full	4.8333	4.6667	4.5	4.8333	4.5	5	4.8333	4.6667
BioRec_SN	4.1667	3.8333	3.8333	3.5	3.3333	4	4	3.5
BioRec_Profile	2.6667	3.5	2.8333	3.6667	2.5	1	3.1667	3
Mirrizi	3.1667	3.5	3	3	2.5	1	2.8333	2.8333
Google	1.5	3	2	2.1667	1.8333	1	2.3333	2.3333

was 0.000000896. Such results suggest that our approach enhances the precision of the recommendations of academic articles in the field of bioinformatics.

Table (4) shows the MAP scores that have been satisfied in each system regarding each assigned task. These tasks were designed to test different functions provided by our recommender system, starting from a general task and gradually moving on to a complicated one. For instance, task 1 was designed to test the utilisation of system BioRec_Full (which was equipped with semantic relation (i.e. sibling relation) as well as an automatic adaptive ontological user profile) and compare it with that of the other systems (BioRec_SN, BioRec_Profile, Mirrizi and Google) in order to provide recommendations on articles that discuss general ideas about bioinformatics, such as definitions, histories, etc., so, the bioinformatician can refresh his/her knowledge of general information about bioinformatics. The remaining tasks took the following form: task 2 was designed to provide recommendations on articles that discuss specific tools which used in bioinformatics such as alignment. Task 3 was created to assess recommender approach in providing recommendations on articles that mentioned programming languages used in bioinformatics such as Perl. Task 4 was designed to provide recommendations on articles about bioinformatics ontologies such as GO or PO. This task is more complex than the previous ones, as it should recommend to each user articles that discuss ontologies previously indicated by the user or relevant articles based on semantic relations that have been identified as part of the interaction with the system. Task 5 was designed to recommend the user with articles that mentioned bioinformatics journals such as BMC this task may help him/her to broaden his/her horizon and be aware of the most important journals in the field.

Figure 2 (i.e. shows the results of task 1) demonstrates that system BioRec_Full outperformed the other systems. Moreover, it shows that system BioRec_SN had a dramatic decrease in comparison with BioRec_Full and also performed less well than systems BioRec_Profile and Mirrizi did, but it was better than the Google system. This suggests that using an automatic adaptive ontological user profile only is not effective in providing recommendations about general topics; however, it is still sufficient in comparison with system Google, which provided standard recommendations. Both systems BioRec_Profile and Mirrizi achieved very close scores in MAP, and this can indicate two main conclusions: (i) semantic enrichment demonstrates an important role in enhancing recommendations even without using a user profile. (ii) The user profile used in system Mirrizi is still weak because it does not use ontologies to represent the user profile and does not apply a method that can keep the user preferences updated. Also, it does not exploit semantic relations successfully and employ them in order to enhance recommendations. However, both systems have good performance in comparison with system Google, which represents the weakest system in this comparison, as it provides standard recommendations.

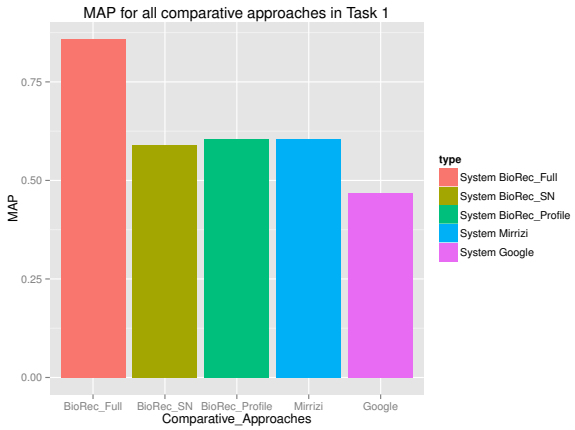

Figure 2: MAP Metric for Task 1.

Finally, another comparison was taken for all comparative approaches for the same tasks. However, this time, users were asked to assess the recommendations based on the specific interests they chose from their profiles. They were asked to assess only the top 10 results; this threshold was chosen so as not to burden the users with assessing the same task again. This evaluation was completed by considering MAP to assess the level of success achieved when selecting specific interests to have recommendations on. Figure 3 shows the result of the systems' performance regarding each assigned task. System BioRec_Full outperformed the other systems. Then, the performance gradually decreased when moving on to another system until reaching system Google. This supports our hypotheses that considering both semantic relations (i.e. sibling) and the adaptive ontological user profile can contribute to enhancing the accuracy of the provided recommendations.

6. CONCLUSION

To sum up, this paper discussed a semantic reasoning method that exploited sibling semantic relations occurring as a result of information overlapping between different bioinformatics resources in order to enhance the recommendations provided in bioinformatics articles. This method showed an enhancement in the accuracy of the provided recommendations when compared with an approach from the literature that uses a content-based filtering method and a user profile to provide recommendations. However, the literature approach did not exploit the semantic information through multiple resources successfully. Moreover, our method also compared favourably to methods using only the user profile or only the semantic relation (i.e. sibling). In addition,

Table 4: MAP for all assigned tasks in sibling enrichment comparison

Systems/Tasks	Task 1	Task 2	Task 3	Task 4	Task 5
BioRec_Full	0.8594	0.7733	0.8054	0.8917	0.8946
BioRec_SN	0.5890	0.7017	0.6620	**0.7619**	**0.7833**
BioRec_Profile	**0.6044**	0.5797	0.6093	**0.6459**	**0.4977**
Mirrizi	**0.6048**	0.5326	**0.5305**	**0.5751**	**0.4743**
Google	**0.4674**	**0.4419**	**0.4077**	**0.5735**	**0.4770**

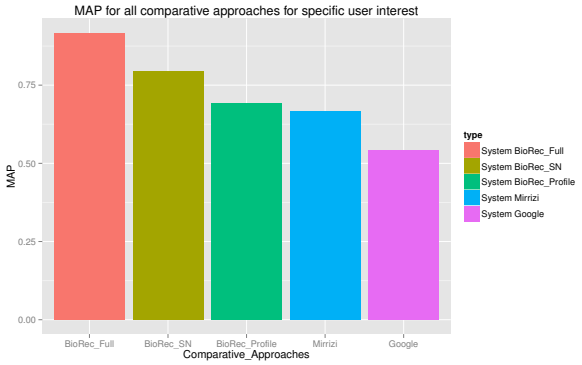

Figure 3: MAP for All Comparative Approaches for Specific User Interest.

it outperformed the baseline, as the baseline did not consider the semantic relations when providing recommendations. Furthermore, our recommender approach can provide more sufficient results for specialist search, where the more information can be utilised on the recommendations, the more accurate the results that can be gained.

Currently, we are developing a new method that exploits semantic similarities between different concepts during the formulation of the semantic network. As the semantic network represents the overlapping/complementarity information between different resources, then semantically similar concepts will be exploited to enhance the accuracy of the provided recommendations in the field of bioinformatics. This method will be compared with the current method, literature method and baseline in order to assess the level of success achieved when applying such a method to our recommender approach.

7. REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Publishing Company, USA, 2nd edition, 2008.

[2] T. Bogers and A. van den Bosch. Recommending scientific articles using citeulike. In *RecSys'08*, pages 287–290, 2008.

[3] P. Borlund. The iir evaluation model: a framework for evaluation of interactive information retrieval systems. *Information research*, 8(3), 2003.

[4] T. Di Noia, R. Mirizzi, V. C. Ostuni, D. Romito, and M. Zanker. Linked open data to support content-based recommender systems. In *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS'12, pages 1–8. ACM, 2012.

[5] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[6] L. Ding, P. Kolari, Z. Ding, and S. Avancha. Using ontologies in the semantic web: A survey. In *Ontologies*, pages 79–113. Springer, 2007.

[7] S. Dooms, T. De Pessemier, and L. Martens. A user-centric evaluation of recommender algorithms for an event recommendation system. In *Workshop on Human Decision Making in RecSys'11*, pages 67–73, 2011.

[8] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Mymedialite: A free recommender system library. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 305–308. ACM, 2011.

[9] B. P. Knijnenburg, M. C. Willemsen, and A. Kobsa. A pragmatic procedure to support the user-centric evaluation of recommender systems. In *RecSys'11*, pages 321–324. ACM, 2011.

[10] C. Liu, R. W. White, and S. Dumais. Understanding web browsing behaviors through weibull analysis of dwell time. In *Proceedings of SIGIR'10*, pages 379–386. ACM, 2010.

[11] D. L. McGuinness, F. Van Harmelen, et al. OWL web ontology language overview. *W3C recommendation*, 10(2004-03), 2004.

[12] M. Mezghani, C. A. Zayani, I. Amous, and F. Gargouri. A user profile modelling using social annotations: a survey. In *WWW'12*, pages 969–976, 2012.

[13] S. E. Middleton, D. De Roure, and N. R. Shadbolt. Ontology-based recommender systems. In *Handbook on ontologies*, pages 779–796. Springer, 2009.

[14] E. Miller. An introduction to the resource description framework. *Bulletin of the American Society for Information Science and Technology*, 1998.

[15] R. Mirizzi, T. Di Noia, V. C. Ostuni, and A. Ragone. Linked open data for content-based recommender systems. Technical report, http://sisinflab.poliba.it/semantic-expert-finding/papers/tech-report-1-2012.pdf, 2012.

[16] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[17] H. Stuckenschmidt and A. Schlicht. Structure-based partitioning of large ontologies. In *Modular Ontologies*, pages 187–210. Springer, 2009.

[18] L. Tari, N. Vo, S. Liang, J. Patel, C. Baral, and J. Cai. Identifying novel drug indications through automated reasoning. *PloS one*, 7(7):e40946, 2012.

[19] J. Trajkova and S. Gauch. *Improving ontology-based user profiles*. PhD thesis, University of Kansas, Electrical Engineering and Computer Science, 2003.

[20] T. Yoneya and H. Mamitsuka. Pure: a pubmed article recommendation system based on content-based filtering. In *Genome informatics.*, volume 18, pages 267–276, 2007.