Allied: A Framework for Executing Linked Data- Based Recommendation Algorithms

Cristhian Figueroa, Iacopo Vagliano, Oscar Rodríguez Rocha, Marco Torchiano, Catherine Faron Zucker, Juan Carlos Corrales, Maurizio Morisio

Preprint of the paper publised on the IJSWIS Journal

Allied: A Framework for Executing Linked Data-based Recommendation Algorithms

Cristhian Figueroa, Politecnico di Torino, Turin, Italy and Universidad del Cauca, Popayán, Colombia.

Iacopo Vagliano, Politecnico di Torino, Turin, Italy.

Oscar Rodríguez Rocha, INRIA, Sophia Antipolis, France.

Marco Torchiano, Politecnico di Torino, Turin, Italy.

Catherine Faron-Zucker, I3S, Université Nice Sophia Antipolis - CNRS, Sophia Antipolis, France.

Juan Carlos Corrales, Universidad del Cauca, Popayán, Colombia.

Maurizio Morisio, Politecnico di Torino, Turin, Italy.

ABSTRACT

The increase in the amount of structured data published on the Web using the principles of Linked Data means that now it is more likely to find resources on the Web of Data that represent real life concepts. Discovering and recommending resources on the Web of Data related to a given resource is still an open research area. This work presents a framework to deploy and execute Linked Data based recommendation algorithms to measure their accuracy and performance in different contexts. Moreover, application developers can use this framework as the main component for recommendation in various domains. Finally, this paper describes a new recommendation algorithm that adapts its behavior dynamically based on the features of the Linked Data dataset used. The results of a user study show that the algorithm proposed in this paper has better accuracy and novelty than other state-of-the-art algorithms for Linked Data.

Keywords: Linked Data, Recommender System, Semantic Recommender, Web of data, Evaluation Framework, Recommender Algorithm, DBpedia, Interlinked Data

INTRODUCTION

Due to the increase in the amount of structured data published on the Web through the principles of Linked Data, it is more likely to find resources that describe or represent real life concepts. The information provided by these resources can be used in different domains. However, finding and recommending related resources is still an open research area (Ricci, Rokach & Shapira, 2011). A Systematic Literature Review (Figueroa, Vagliano, Rodríguez Rocha & Morisio, 2015) stated that the problem of

finding existing relationships between resources can be addressed by analyzing the categories they belong to, their explicit references to other resources and / or by combining both of these approaches. Currently there are many works aimed at resolving this problem by focusing on specific application domains and datasets.

In this context, the present work aims to answer the following research questions:

- How can we choose the best existing algorithm for recommending resources from the Web of Data, which best suits the characteristics of a given application domain and a given dataset?
- How can we measure the performance and accuracy of the different existing algorithms to select the one that best suits specific recommendation needs?
- Is it possible to have an algorithm that is dynamically adaptable to the characteristics of the dataset and independent of the application domain?

To answer these research questions, the authors propose a framework for deploying and executing Linked Data based recommendation algorithms (implemented following some guidelines), to facilitate the conduction of studies to evaluate them in different application domains and without being bounded to a single dataset. Thus, the framework makes it possible to benchmark the algorithms to choose the one that best fits the recommendation requirements.

Additionally, the framework provides a set of APIs that enable application developers to use it as the main component for recommendation in various contexts. In this way, developers do not need to deal with the execution platform of the algorithms but only to focus their efforts either on selecting an existing algorithm or on writing a customized one.

By using the previously mentioned framework and after conducting a deep analysis of the behaviors and benchmark results of state-of-the-art recommendation algorithms, the authors created a new recommendation algorithm that adapts dynamically to the characteristics of the dataset and to the application domain on which it is used.

The remainder of this paper is structured as follows: section *Related Work* presents the state of the art about Linked Data based Recommender Systems (RS). Section Research Methodology describe the research approach and outlines the main steps undertook. Section *The Framework* introduces an evaluation framework for deploying recommendation algorithms. Section *Implementation* details the framework including the main modules for discovering, ranking and categorizing resources. Section *A Dynamic Algorithm for Recommendation* proposes a new algorithm: *ReDyAl*. Section *Evaluation* describes a user study conducted to evaluate *ReDyAl* in comparison with state-of-the-art algorithms based on Linked Data. Finally, section *Conclusions and Future Work* presents the conclusions and future work.

RELATED WORK

The research work presented in this paper is based on the results obtained from a systematic literature review (Figueroa, et al., 2015). Therefore, in this paper the RS were classified in the following types:

Graph-based. This is the most common type of algorithms used in RS based on Linked Data. These algorithms exploit the graph structure of datasets for computing relevance scores for items represented as nodes in a graph. Algorithms in this category are classified into:

- *Semantic Exploration:* explore the graph structure of datasets using structural relationships to compute distances for recommendations. HyProximity (Damljanovic, Stankovic, & Laublet, 2012); dbrec (Passant, 2010; Kitaya, Huang, & Kawagoe, 2012); page rank (Musto, Basile, Lops, De Gemmis, & Semeraro, 2014; Nguyen, Tomeo, Di Noia, & Di Sciascio, 2015); semantic clustering (H. G. Ko, Kim, Ko, & Chang, 2014), and VSM (Musto et al., 2014).
- *Path-based:* use information about semantic paths within a RDF graph structure to compute distances. Spreading activation (Marie, Gandon, Legrand, & Ribi ere, 2013; Hajra et al., 2014; Cheekula, Kapanipathi, Doran, & Jain, 2015); random walk (Cantador, Konstas, & Jose, 2011); and path-weights for vertex discovery (Strobin & Niewiadomski, 2014).

Table 1, shows a comparison between graph-based algorithms.

Table 1: Comparison of graph-based algorithms

Advantages	Disadvantages
• Serendipitous recommendations. • Explanations of recommendations following paths. • Creation of domain-independent RS. • Exploiting hierarchical information to categorize recommendations.	• High cost of exploiting semantic features due to inconsistency of datasets. • No contextual information. • High computational complexity for large datasets. • Need for dataset customization to address the computational complexity.

Machine Learning. This is the second most common type of algorithms used in RS based on Linked Data. This type of algorithms uses techniques from data mining in order to analyze, predict and classify data extracted from Linked Data datasets to produce recommendations. Algorithms in this type are classified into:

- *Supervised:* a model is prepared through a training process where it produces predictions about class labels from attributes. kNN (Ristoski, Mencía, & Paulheim, 2014), decision trees (Ostuni, Di Noia, Di Sciascio, & Mirizzi, 2013; Ristoski et al., 2014); logistic regression (Ostuni et al., 2013; Moreno et al., 2014; Musto et al., 2014); SVM (Kushwaha & Vyas, 2014; Ostuni, Di Noia, Mirizzi, Di Sciascio, & Noia, 2014; Khrouf & Troncy, 2013); random forest (Ostuni et al., 2013; Musto et al., 2014); and naive Bayes (Schmachtenberg, Strufe, & Paulheim, 2014).
- *Unsupervised:* input data is not labelled and does not have a known result, so they aim to discover the structure or distribution of the data. K-Means (Moreno et al., 2014; Manoj Kumar, Anusha, & Santhi Sree, 2015); fuzzy-C means, SOM and PCA (Ostuni et al., 2014).

Table 2 shows a comparison between machine learning algorithms.

Table 2: Comparison of machine learning algorithms

Advantages	Disadvantages
• Many algorithms are already developed for recommendations. • Some algorithms can deal with large datasets in a reasonable execution time. • Algorithms may be configured to automatically improve their results with experience.	• Time-consuming algorithms for training phase. • Most of the RS use Linked Data to enrich data of items or users, so the intrinsic semantic structure of the Linked Data is not considered.

Memory-based. Algorithms for rating predictions based on the entire collection of previously rated path queries. Rating prediction (Kushwaha & Vyas, 2014; Moreno et al., 2014; Musto et al., 2014); SVD (Moreno et al., 2014; Ko, H. G, Son, J., & Ko, I.Y. 2015); and matrix factorization (Lommatzsch, Kille, & Albayrak, 2013). Table 3 shows a comparison between memory-based algorithms.

Table 3: Comparison of memory-based algorithms

Advantages	Disadvantages
• Well established algorithms for RS based mainly on collaborative filtering approaches.) • Easy to implement / use.	• Cold-start problem for users or items. • Time-consuming algorithms

As detailed in section *The Framework*; the *Allied* framework was initially implemented on two fundamental graph-based RS: *HyProximity* presented by Damljanovic et al. (2012) and *dbrec* proposed by Passant (2010). Furthermore, the authors propose a new recommendation algorithm, already implemented in *Allied*, and the framework has enabled to comparatively evaluate these three implemented algorithms. This work focuses on algorithms that rely only on Linked Data, but it is not limited to them, since the framework is designed for being extended to consider other approaches in combination with Linked Data.

RESEARCH METHODOLOGY

To addressed the research questions previously introduced, the research conducted followed the steps showed in Figure 1. First, a systematic literature review was conducted. Then, the Allied framework was developed. Afterwards, ReDyAl was proposed and integrated into Allied. It is a new dynamic algorithm for concept recommendation based on Linked Data. Furthermore, a user study was conducted to evaluate the relevance and novelty of proposed algorithm.

Figure 1: The research plan followed

The systematic literature review focuses on research problems addressed and on the contributions proposed in Linked Data based RS. The work described in this thesis is based on the outcome of this review. The work includes several research aspects. For this reason, it was the result of a collaboration among various participants, which provided their experience in different research areas: The SoftEng research group (Politecnico di Torino) addressed tasks related with Software Engineering, the WIMMICS group (INRIA, I3S, Université Nice Sophia Antipolis - CNRS) provided their experience in Semantic Web, and the Telematics Engineering group (Universidad del Cauca) supported the implementation of applications and services on the Web.

THE FRAMEWORK

Allied[1] is a framework to deploy and execute resource recommendation algorithms based on Linked Data. Through an implementation of these algorithms, it is possible to test them in different application domains and to analyze their behaviors.

Accordingly, the framework facilitates the comparison of the results for these algorithms both in performance and relevance. In this way, the framework creates an environment to select, evaluate, and develop algorithms to recommend resources belonging to different contexts and application domains that can be executed within the same environment and with different configuration parameters. In addition, it enables the creation of innovative applications on top of it.

For studying recommendation algorithms, the recommendation process has been divided into four steps (as shown in Figure 2):

Figure 2: Steps of the recommendation process

Resource generation. The first step is intended to generate a set of candidate resources (CR) that maintain semantic relationships with an initial resource (ir). The semantic relationships are direct or indirect links between two resources in a Linked Data dataset.

Results ranking. It sorts the candidate resources generated in the previous step by considering the semantic similarity with the initial resource. In this step, different semantic similarity measures can be used to calculate the semantic similarity between pairs of resources.

Scope-based categorization. The list of ranked candidate resources generated in the previous step may be too general, that is, a recommendation may include resources from unrelated domains of knowledge. For this reason, this optional third step groups these resources already ranked into meaningful clusters that represent common knowledge domains.

Results presentation. Finally, the results of the last step are graphically presented through different facets to allow the end-users to visualize the recommendations.

IMPLEMENTATION

The algorithms implemented for each layer of *Allied* are shown in Figure 3. *Generation*, *Ranking*, and *Classification* layers are responsible for the recommendation process, while *Knowledge base core* and *Presentation* providing the access to the datasets and presenting the results.

Figure 3: Diagram of the implementation of Allied

Knowledge base core

This module represents the data layer of the *Allied* framework. It is the main data source containing knowledge about resources and their structural relationships. The knowledge base may be seen as a tuple (R, T, L) composed by resources (R), categories (T), and relationships (L), where:

- *Resources* are abstractions from the real life like ideas or notions[2].
- *Categories* are the bases of the class hierarchy for the knowledge items. DBpedia provides information about the hierarchical relationships in three different classification schemata: Wikipedia Categories, YAGO Categories[3], and WordNet Synset Links[4]. In this implementation, the Wikipedia categories that are represented with concepts of the Simple Knowledge Organization System (*SKOS*[5]) vocabulary were chosen to describe categories and their relationships.
- *Relationships* are the links (also known as properties) connecting resources or categories along the whole dataset graph. The knowledge base for the framework contains three types of relationships.
  - Resource-Resource: these are the traversal relationships between resources, which are those links between resources that do not refer to hierarchical classifications. Most of the links of DBpedia belong to this type.
  - Resource-Category: these are relationships between a resource and a category. They can be represented by using the SKOS properties `skos:subject` (hasCategory) and `skos:isSubjectOf` (IsCategoryOf). However, `skos:subject` and `skos:isSubjectOf` are deprecated (Miles & Bechhofer, 2009) and consequently not used in DBpedia. Therefore, DBpedia relates resources to their Wikipedia categories using `dcterms:subject` instead. Accordingly, `dcterms:subject` is used in *Allied* for both relationships.
  - Category-Category: these are hierarchical relationships between categories within a hyponymy structure (a category tree). They can be represented by using the SKOS properties `skos:broader` (isSubCategoryOf) and `skos:narrower` (isSuperCategoryOf).

The current implementation of *Allied* uses the DBpedia dataset as knowledge base, but it can be easily extended to other datasets. DBpedia was selected because it is a general dataset that offers the possibility to evaluate the results in many scenarios. DBpedia is one of the biggest datasets that is frequently updated because its data comes from Wikipedia, and that continuously grows into one of the most interlinked datasets in the

Web of Data (Schmachtenberg, Bizer & Paulheim, 2014). The Wikipedia categories (SKOS concepts) were selected because they are the most linked in DBpedia[6].

Generation layer

This layer aims at discovering resources related to a given one through semantic relationships. Given an initial resource (or a set of initial ones) it generates a set of candidate resources located at a predefined distance. For this layer, three generators were implemented based on the semantic relationships found on the Linked Data: (i) a traversal generator in order to study direct and indirect relationships between resources (Resource-Resource) avoiding hierarchical relationships; (ii) a hierarchical generator for indirect relationships between resources through direct relationships between resources and categories (Resource-Category) and between categories (Category-Category); and (iii) a dynamic generator which combines dynamically both types of relationships giving priority to the existing interlinking between resources.

Traversal generator

The Traversal generator looks for resources that are directly related to a given initial resource and those found through a third resource (indirect relationships). Its implementation is inspired by the *dbrec* recommender (Passant, 2010).

SPARQL queries are used to retrieve the resources directly and indirectly connected with the initial resource A set of forbidden links can be defined to prevent the algorithm to obtain resources over links pointing to empty nodes (i.e. resources without a URI), literals that are used to identify values such as numbers and dates or nodes that are not desired for the recommendation. In other words, it is a way to limit the results of the algorithm.

Hierarchical generator

The hierarchical generator generates a set of candidate resources located at a specified distance in a hierarchy of categories taken from a category tree described in a dataset. The implementation of this module is inspired by the work of Damljanovic et al. (2012), which obtains candidate resources by navigating a category tree of the Wikipedia categories.

The hierarchical generator firstly extracts base categories of an initial resource (`<inURI>`) and then looks for broader categories until a maximum distance (which may be user-defined) is reached. This maximum distance is the hierarchical distance of a broader category from base categories. It is inversely proportional to the level of specificity of a category (i.e. the higher the distance the lower the level of specificity of a category).

After extracting categories, this module extracts subcategories for all the broader categories at maximum distance (i.e. it descends one level into the category tree) to increase the possibility for finding more candidate resources. Finally, the algorithm obtains candidate resources for each category (including sub-categories).

.

Thus, the module creates a “category graph”, including the initial resource, its category tree, and the candidate resources retrieved for each category. For example, Figure 4

shows an example of the category graph for the resource `<http://dbpedia.org/resource/Mole_Antonelliana>`.

Figure 4: Example of a category graph for the resource "Mole Antonelliana" (candidate resources are not included for space reasons).

Dynamic generator

The Dynamic generator is a "hybrid" generator, which takes advantage of both the traversal and the hierarchical approaches, giving priority to the existing interlinking between resources, that is, one of the four principles of Linked Data (Bizer, Heath & Berners-Lee, 2009). The innovative algorithm of this generator is explained in section *A Dynamic Algorithm for Recommendation*.

Ranking layer

This layer mainly ranks candidate resources obtained in the previous layer, based on semantic similarity functions. These candidate resources are sorted by the descendant values of a semantic similarity function, which measures the similarity between the initial resource and each one of these candidate resources. The framework in its current implementation includes (but is not limited to) three ranking algorithms.

Traversal LDSD ranking

The traversal LDSD ranking algorithm calculates the Linked Data Semantic Distance *(LDSD)* between an initial resource and each one of the candidate resources obtained in the generation layer. The LDSD distance, initially proposed by Passant (2010), is based on the number of indirect and direct links between two resources. The similarity of two resources (r_1, r_2) is measured in Equation (1), which is the basic form of the *LDSD* distance.

$$LDSD(r_1, r_2) = \frac{1}{1+Cd_{out}+Cd_{in}+Ci_{out}+Ci_{in}} \qquad (1)$$

Cd_{out} is the number of direct input links (from r_1 to r_2), Cd_{in} is the number of direct output links, Ci_{in} is the number of indirect input links, and Ci_{out} is the number of indirect output links.

Unlike the implementation developed by Passant, which is limited to links from a specific domain, the LDSD function implemented in *Allied* considers all resources from the dataset. However, it can be customized to defined types of links belonging or not to a specific domain by adding a set of *forbidden links*.

Two SPARQL queries counts direct and indirect input and output links between an initial resource and a resource of the set of candidate resources The traversal ranking algorithm calculates the LDSD for each pair of resources composed of an initial resource and each of the resources obtained from the generation layer.

HyProximity ranking

The HyProximity ranking algorithm is based on the similarity measure defined by Stankovic, Breitfuss & Laublet (2011). This measure can be used to calculate both traversal and hierarchical similarities. The *HyProximity* in its general form is shown in Equation (2) as the inverted distance between two resources, balanced with a pondering function.

$$hyP(r_1, r_2) = \frac{p(r_1,r_2)}{d(r_1,r_2)} \quad (2)$$

In this equation d is the distance function between two resources and p is a weight function used to give a level of importance to different distances. Based on the structural relationships (hierarchical and traversal), different distances and pondering functions may be used to calculate the *HyProximity* similarity.

- *Hierarchical HyProximity*: The definition of this similarity function relies on the work of Stankovic, Breitfuss & Laublet (2011). It was calculated using the maximum distance of categories of the hierarchical generator algorithm such that $d(ir, cr_i) = maxLevel$, ir is the initial resource and cr_i is each one of the candidate resources generated in the hierarchical algorithm. The pondering function is defined in Equation (3), which is an adaptation of the informational content function defined by Seco, Veale, & Hayes (2004). In this equation, $hypo(C)$ is the number of descendants of category C and $|C|$ is the total number of categories in the categoryGraph of C.

  $$p(C) = 1 - \frac{\log(hypo(C)+1)}{log(|C|)} \qquad (3)$$

  This function was selected because it minimizes the complexity of calculation of the informational content with regard to other functions that employ an external corpus (Hadj Taieb, Ben Aouicha, Tmar, & Hamadou, 2011).

- *Traversal HyProximity*: in this similarity function $d(ir, cr_i) = maxLevel$ if the generator of resources is hierarchical, otherwise $d(ir, cr_i) = 1$ for resources connected to the initial resource through direct traversal links or $d(ir, cr_i) = 2$ for indirect traversal links. The pondering function is defined in Equation (4): $p_{trav}(r_1, r_2)$ depends on the number of resources n connected over a specific property and the total number of resources of the dataset M:

  $$p_{trav}(r_1, r_2) = -\log\frac{n}{M} \quad (4)$$

  Nonetheless, in *Allied*, this algorithm is not limited to a specific property, and optionally can be configured to support a set of forbidden links or allowed links in a similar way as shown in the *Generation Layer*. The number of direct and indirect links was calculated with SPARQL queries. The value of M was fixed to the number of resources contained in DBpedia.

Classification layer

Since this implementation of the framework is based on DBpedia, which is a general-purpose dataset, the results obtained may contain an inherent ambiguity due to the generality of the data used to produce recommendations. Moreover, a single ranked list

of recommendations may not always be a good way to show this kind of general results because users may require results arranged with their subjective needs or knowledge domain. To satisfy this requirement, the classification layer provides mechanisms to group the results obtained from the ranking layer into meaningful clusters that represent domains of knowledge.

Currently, the classification layer relies on Algorithm 1The classification algorithm provides a mechanism to easily access the recommended items organized by clusters. Although, in the current implementation of *Allied* the resulting clusters correspond to Wikipedia categories, it is easy suitable to define custom clusters by aggregating many categories or to rely on other category schemas from the Web of Data, such as YAGO categories.

Algorithm 1. Hierarchical classification algorithm

```
Require: CR, inURI, maxLevel, optionally Gc_in
Ensure: A Category graph G_C
1:    if Gc_in = null then
2:        Gc = createCategoryGraph(CR, maxLevel)
3:    Else
4:        Gc = Gc_in
5:    end if
6:    C_maxLevel = getMaxLevelCategories(Gc)
7:    for each pair of categories (c_i, c_j) ∈ C_maxLevel do
8:        c_lcb = getLesCommonBroaderCategory(c_i, c_j)
9:        Add c_lcb to Gc
10:       Add edge (c_i, c_lcb) and edge (c_j, c_lcb) to Gc
11:   end for
12:   intersectCategories(Gc)
13:   deleteEmptyCategories(Gc)
13:   return Gc
```

Algorithm 1 receives as input a set of ranked candidate resources (CR), an initial resource $inURI$, and optionally an initial category graph (Gc_{in}) (in case that a hierarchical structure is already available). If Gc_{in} is not given, then the algorithm creates a new category graph Gc containing categories for the initial resource and the set of candidate resources until a maximum distance ($maxLevel$) (Lines 1 - 5). In this implementation $maxLevel$ is set to 2 because with this value it is possible to obtain a reasonable relationship between the number of categories and the time consumed.

Afterwards, the algorithm extracts categories at the highest distance ($C_{maxLevel}$) and creates pairs of categories combining the elements of $C_{maxLevel}$ (Lines 6 - 7). Next, the function *getLessCommonBroaderCategory*, which is based on the less common ancestor, is executed to find a set of broader categories subsuming the categories of $C_{maxLevel}$. These new broader categories are then added to Gc including their edges ((c_i, c_{lcb}) and (c_j, c_{lcb})) (Lines 8 - 11).

Finally, the updated set of categories of Gc are intersected and a function *deleteEmptyCategories* is executed to remove from the graph those categories subsuming less than three subcategories (i.e. only categories c_i, c_j). In this way a classification of higher distance for the candidate resources is created (Lines 12 - 13).

Presentation layer

Allied can easily be integrated to any application that requires recommendations based on Linked Data. The current implementation includes three main interfaces that provide mechanisms to present results to the final user: a Web interface, a standalone interface and a RESTful interface.

A DYNAMIC ALGORITHM FOR RECOMMENDATION

ReDyAl is an algorithm that was developed considering the different types of relationships between data published under the Linked Data principles. It aims to discover related resources from datasets that may contain either "well-linked" resources as well as "poor-linked" resources. A resource is said to be "well-linked" if it has many links higher than the average number of links in the dataset; otherwise it is "poor-linked". The algorithm can dynamically adapt its behavior to find a set of candidate resources to be recommended, giving priority to the implicit knowledge contained in the Linked Data relationships.

ReDyAl can be divided into three stages:

- The first stage discovers resources by analyzing the links or relations (interlinking) between the given initial resource and other resources.
- The second stage analyzes the categorization of the given initial resource and discovers similar resources located in common categories.
- The last stage intersects the results of both the previous stages, given priority to those found in the first stage.

Additionally, the algorithm may be configured with a set of forbidden links to restrict the kind of links the algorithm should consider. For example, DBpedia is a generic dataset containing millions of links between resources, and if a developer is creating an application in the music domain then he/she may be interested only in resources of that domain, so he/she may want to consider only links pointing to those resources.

Figure 5: Flowchart of ReDyAl

Figure 5 shows a flowchart of the *ReDyAl* algorithm, which receives as input an initial resource by specifying its corresponding URI ($inURI$), and three values ($minT$, $minC$, $maxDistance$) for configuring its execution. $minT$ is the minimum number of links (or triples involving the initial resource) to consider a resource as "well-linked". If the initial resource is "well linked", traversal interlinking has a higher priority in the generation of candidate resources, otherwise the algorithm gives priority to the hierarchical relationships. $minC$ is the minimum number of candidate resources that the algorithm is expected to generate, while $maxDistance$ limits the distance (number of

hierarchical levels) that the algorithm considers in the category tree. The value of $maxDistance$ may be defined manually and it is useful when there are not enough candidate resources from the categories found at a certain distance (i.e. the number of candidate resources retrieved is lower than *minC*). In this case, the algorithm increases the distances in order to find more resources and if the *maxDistance* value is reached with less than *minC* candidate resources , the algorithm ranks only the candidate resources found until that moment. Additionally, the algorithm may receive a list of "forbidden links" (*FL*) to avoid searching for candidate resources over a predefined list of undesired links.

Algorithm 2. ReDyAl algorithm

```
Require: inURI, minT, minC, FL, maxDistance
Ensure: A set of candidate resources CR
1:    L_in = readAllowedLinks(inURI, FP)
2:    if |L_in| ≥ minT then
3:       for all l_k ∈ L_in do
4:          DR_{l_k} = getDirectResources(l_k)
5:          IR_{l_k} = getIndirectResources(l_k)
6:          Add DR_{p_k} to CR_tr
7:          Add CR_{p_k} to CR_tr
8:       end for
9:       if |CR_tr| ≥ minC then
10:         return CR_tr
11:      Else
12:         currentDistance = 1
13:         Gc = createCategoryGraph(inURI, currentDistance)
14:         while currentDistance ≤ maxDistance do
15:            CR_hi = getCandidateResources(Gc)
16:            if |CR_hi| ≥ minC then
17                Add CR_tr and CR_hi to CR
18:               return CR
19:            end if
20:            increase currentLevel
21:            updateCategoryGraph(currentLevel)
22:         end while
23:         Add CR_tr and CR_hi to CR
24:      end if
25:   end if
26:   return CR
```

ReDyAl (Algorithm 2) starts by retrieving a list of allowed links from the initial resource. Allowed links are those that are not specified as forbidden (FP) and that are explicitly defined in the initial resource (described in its RDF file). If there is a considerable number of allowed links, i.e., the initial resource is well-linked, the

algorithm obtains a set of candidate resources located through direct (DR_{l_k}) or indirect links (IR_{l_k}) starting from the links explicitly defined in the RDF of the initial resource (Lines 1-8). Next, the algorithm counts the number of candidate resources generated until this point (CR_{tr}) and if it is greater than or equal to $minC$, the execution terminates returning the results (Lines 9-10). Otherwise, the algorithm generates a category graph (Gc) with categories of the first distance and applies iterative updates over the category graph over n distances above the initial resource obtaining broader categories until at least one of two conditions is fulfilled: the number of candidate resources is enough ($CR > minC$), or the maximum distance is reached ($currentDistance > maxDistance$). At each iteration, candidate resources (CR_{hi}) are extracted from the broader categories of maximum distance (Lines 14- 23). In any case the algorithm combines these results with the results obtained in Lines 3 – 8 (Adding CR_{tr} and CR_{hi} to CR). Finally, the set CR of candidate results is returned (Line 26).

EVALUATION

Allied enables the comparative evaluation of any new algorithm with respect to the state-of-the-art since it is possible to deploy all the algorithms to be compared in the same environment. Accordingly, the accuracy and the novelty of the *ReDyAl* algorithm were evaluated using *Allied*. This evaluation aimed to answer the following questions:

RQ1: Which of the considered algorithms is more accurate?

RQ2. Which of the considered algorithms provides the highest number of novel recommendations?

As already mentioned in section *Related Work*, this paper is focused on the comparison of *ReDyAl* with algorithms that rely exclusively on Linked Data to produce recommendations.

Experiment

A user study was conducted involving 109 participants. The participants were mainly students of Politecnico di Torino (Italy) and University of Cauca (Colombia) enrolled in IT courses. The average age of the participants was 24 years old and they were 91 males, 14 females, and 4 of them did not provide any information about their sex. Although the proposed algorithm is not bounded to any domain, this evaluation was focused on movies because in this domain a quite large amount of data is available on DBpedia. Additionally, finding participants was not too difficult, since no specific skills were required to be able to express an opinion about movies. The evaluation was conducted as follows.

Lists of the 20 more representative movies for each initial movie were created. These lists were generated by merging the top 10 movies in the recommendations that each algorithm generated for a given initial movie. Then, the lists of 20 movies were delivered to the users so that they could evaluate the relevance of these recommendations for each initial movie. For each recommendation two questions were asked:

Q1: Did you already know this recommendation? Possible answers were yes, yes but I haven't seen it (if it is a movie) and no.

Q2: Is it related to the movie you have chosen? Possible answers were I strongly agree, I agree, I don't know, I disagree, I strongly disagree. Each answer was assigned respectively a score from 5 to 1.

The authors developed a website[7] to collect the answers from the participants. The participants could choose an initial movie from a list of 45 movies selected from the IMDB[8] top 250 list. The first 50 movies were considered and 5 movies were excluded because they were not available in DBpedia. When a participant selected an initial movie the tool provided the corresponding list of recommendations with the questions mentioned above. Each participant could evaluate recommendations from as many initial movies as he wanted. Therefore, the recommendations of the lists for 40 out of 45 initial movies were evaluated by at least one participant and each movie was evaluated by an average of 6.18 participants. The dataset with the initial movies and the lists of recommendations is available online[9].

With regard to the questions stated at the beginning of this section: RQ1 was satisfied with the Root Mean Squared Error (RMSE) (Shani & Gunawardana, 2011) and RQ2 with the ratio between the number of evaluations in which the recommended item was not known by the participants and the total number of evaluations. The scores that the participants gave to the films were considered as reference for the RMSE measure. Then, these scores were normalized within the interval [0,1], and compared with the similarities that each algorithm computed. This is because, each algorithm ranked the candidate movies based on a semantic similarity function.

Results

The results of the evaluation are summarized in Figure 6, which compares the algorithms with respect to their RMSE and novelty. The "sweet spot" area represents the conditions in which an algorithm has a good trade-off between novelty and prediction accuracy. In effect, presenting a high number of recommendations not known to the user is not necessarily good because it may prevent him to assess the quality of the recommendations: for example having in the provided recommendation a movie which he has seen and which he liked may increase the trust of the user in the RS.

With regard to the RQ1, *HyProximity* accounted the lowest RMSE measures (with 25% and about 36% for the hierarchical and traversal versions respectively). Though, these results are less significant due to the low number of answers to Q2 for these algorithms (this means that the RMSE was computed over a low number of recommendations). For both *ReDyAl* and *dbrec* the RMSE is roughly 45%. Concerning RQ2, the two versions of *HyProximity* account for the highest values (hierarchical roughly 99%, while traversal about 97%). The high values of novelty means that the algorithm can recommend more novel objects that have not been noticed by the user before, however these low values in performance scored by *HyProximity* hierarchical and traversal imply that most of these novel results are not relevant. In this regard, *ReDyAl* and *dbrec* scored good values for novelty accounting respectively for about 60% and 45%. while presenting also good values for performance.

Table 4: Percentage of answers for Q1 by algorithm

Algorithm	Yes	Yes but I haven't seen it (if it is a movie)	No
ReDyAl	27.95	9.17	62.88
dbrec	41.10	11.95	46.95
HyProximity hierarchical	1.08	0.36	98.56
HyProximity traversal	1.32	1.89	96.79

Figure 6: Accuracy and novelty of the algorithms

HyProximity generated recommendations based in both traversal and hierarchical algorithms, which only obtained few answers to Q2. In this regard, Table 4 shows that most of the recommendations generated were unknown to the users. As a consequence, the results for both algorithms are less definitive than for the other algorithms. This is especially meaningful for RQ1, since only the evaluations for which the answer to Q1 was either *yes* or *yes but I haven't seen it (if it is a movie)* were considered for computing the accuracy measures.

Furthermore, the Fleiss' kappa measure was evaluated for assessing the agreement of the participants answering Q2. The recommendations that were not evaluated by at least one participant were excluded. The scored value for the Fleiss' kappa was 0.79; which corresponds to a substantial agreement (Landis & Koch, 1977).

Finally, Figure 6 shows that *ReDyAl* and *dbrec* provide a good trade off among accuracy and novelty (sweet spot area), although *ReDyAl* performs better in novelty. *HyProximity hierarchical* and *HyProximity traversal* seems to be excellent performers since their RMSE is low and their novelty is high. However, it should be noticed that RMSE was computed on few evaluations. A further analysis of these two algorithms is needed to verify if the user can benefit from such a high novelty and if novel recommendations are relevant. In addition, more research is needed on poorly-linked resources, since the choice of the initial films focused on selecting well known films could ease the evaluation from participants. On poorly-linked resources *ReDyAl* and *Hyproximity* hierarchical are expected to score good values of the accuracy of the recommendations, since they can rely on categories, while *dbrec* and *HyProximity* traversal are likely to provide much less recommendations since they only rely on direct links between resources.

CONCLUSIONS AND FUTURE WORK

PhD Contributions

The main contributions of the research presented in this paper are:

- The first Systematic Literature Review on Recommender Systems based on Linked Data.
- A framework to analyze the results of different recommendation algorithms.
- A dynamic algorithm for concept recommendation based on the knowledge of Linked Data relationships.

- A classification algorithm to categorize the recommendations arranging the candidate concepts into meaningful clusters or contexts.
- A comparative study of the algorithms for RS based on Linked Data.

Conclusions

This paper presented *Allied*, a framework for deploying and executing recommendation algorithms that use Linked Data as their knowledge base. Additionally, *ReDyAl* an hybrid algorithm that dynamically integrates both the traversal and hierarchical approaches for discovering resources is presented. It was designed based on the analysis of state-of-the-art recommendation algorithms and by using *Allied*.

The current version of *Allied* implements a set of three state-of-the-art traversal and hierarchical algorithms and *ReDyAl*. It enables to select the algorithm which best suits for a domain or an application. In addition, since the approach exploited is general, it is possible to adapt *Allied* to other datasets and to select the algorithm which best fits the characteristic of the dataset.

The algorithms currently implemented with *Allied* were evaluated and compared by conducting a user study relying on *Allied*. This framework facilitated the study, since the algorithms were deployed in the same environment and the generated recommendations were aligned. The study demonstrated that *ReDyAl* improves in the novelty of the results discovered, although the accuracy of the algorithm is not the highest (due to its inherent complexity). The study focused on movies because in this domain a quite large amount of data is available on DBpedia and participants were not need to havespecific skills. However, *Allied* allows repeating the study in any other domain.

Future Work

Future work includes studying the relevance under different domains and improving the accuracy of *ReDyAl* while maintaining its novelty. Another limitation to address in future consists in the lack of personalization. As already mentioned, the algorithms currently implemented within Allied rely exclusively on Linked Data to generate recommendations, and do not provide personalized recommendations. However, they can effectively deal with situations where there are not enough data about the user/items. For example, ReDyAl was integrated in a mobile application developed by Telecom Italia which faced a situation of cold-start problem, i.e., new mobile users without user profile information available for recommendations. In this case, ReDyAl demonstrated to be an optimal solution because it can generate recommendations based solely on the semantic extracted from items and their relationships with concepts from Linked Data datasets. Moreover, the framework is designed to be extended, thus collaborative filtering algorithms can be added in future versions to personalize recommendations.

More in general there are still open problems which require further research. Diversity is a popular topic in content-based recommender systems, which usually suffer from overspecialization. Another issue which is gaining interest is mining microblogging data and text reviews. Opinion mining and sentiment analysis techniques can support recommendation methods that consider the evaluation of aspects of items expressed in text reviews. Extracting information from raw text in the form of Linked Data can ease

its exploitation and the integration. Additionally, Linked Data could also be used to explain recommendations since they encode semantic information. This could be particularly useful when unknown items are proposed: the system should assist the user in the decision process, both to justify the suggestion and provide additional information that allows the user to understand the quality of the recommended item. This could increase the transparency and scrutability of the system, and the user's trust and satisfaction.

This study showed that Linked Data based RS generates new recommendations. This is useful because users do not want to receive recommendations about items they already know about or have previously consumed. Additionally, recommending very popular items, which can be easily discovered may not be enough. For this reason, it is important to propose items that are interesting and unexpected. This is known as serendipity and should be further investigated.

Finally, a closely related research area is exploratory search. It refers to cognitive consuming search tasks such as learning or topic investigation. Exploratory search systems also recommend relevant topics or concepts. An open question not addressed in this work is how to leverage the data semantics richness for successful exploratory search.

PHD PROGRAM

Two PhD programs were involved in the development of this research project: the PhD Course in Telematics Engineering (Universidad del Cauca), and the PhD Course in Computers and Systems Engineering (Politecnico di Torino). The former focus on telematic systems and services oriented to the development of the Information and Communication Technologies (ICT). The latter addressescomputer and systems engineering, with main interest in automation, informatics and operational research.

The PhD advisors are: Prof. Juan Carlos Corrales (Universidad del Cauca), whose research areas are services composition and data analysis; and Prof. Maurizio Morisio (Politecnico di Torino) whose interests lies in finding, applying and evaluating the most suitable techniques and process to produces better software faster.

ACKNOWLEDGMENTS

The first author was supported by a fellowship from COLCIENCIAS and the second author by a fellowship from TIM.

REFERENCES

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5, 1–22. doi:10.4018/jswis.2009081901

Cantador, I., Konstas, I., & Jose, J. M. (2011). Categorising social tags to improve folksonomy-based recommendations. *Web Semantics: Science, Services and Agents on the World Wide Web*, *9*(1), 1–15. doi:10.1016/j.websem.2010.10.001

Cheekula, S. K., Kapanipathi, P., Doran, D., & Jain, P. (2015). Entity Recommendations Using Hierarchical Knowledge Bases. *In Proceedings of the 4th Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data at ESWC2015*, 1–12.

Damljanovic, D., Stankovic, M., & Laublet, P. (2012). Linked Data-Based Concept Recommendation: Comparison of Different Methods in Open Innovation Scenario. In *The Semantic Web: Research and Applications* (pp. 24–38). Heraklion, Crete, Greece: Springer. doi:10.1007/978-3-642-30284-8_9.

Figueroa, C., Vagliano, I., Rodríguez Rocha, O., & Morisio, M. (2015). A systematic literature review of Linked Data-based recommender systems. *Concurrency and Computation: Practice and Experience*, 27(17), (pp. 4659–4684). doi:10.1002/cpe.3449

Hadj Taieb, M. A., Ben Aouicha, M., Tmar, M., & Hamadou, A. B. (2011). New information content metric and nominalization relation for a new WordNet-based method to measure the semantic relatedness. In *The IEEE 10th International Conference on Cybernetic Intelligent Systems (CIS)*. (pp. 51–58). doi:10.1109/CIS.2011.6169134

Hajra, A., Latif, A., & Tochtermann, K. (2014). Retrieving and ranking scientific publications from linked open data repositories (p. 29). In *Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business*.

Kitaya, K., Huang, H.-H., & Kawagoe, K. (2012). Music Curator Recommendations Using Linked Data. In *Second International Conference on the Innovative Computing Technology*. (pp. 337–339). Casablanca: IEEE. doi:10.1109/INTECH.2012.6457799

Ko, H. G, Son, J., & Ko, I.Y. (2015). Multi-Aspect Collaborative Filtering based on Linked Data for Personalized Recommendation. *In Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion* (Vol. New Trends, pp. 49–50). New York, USA: ACM Press. doi:10.1145/2740908.2742780

Ko, H. G., Kim, E., Ko, I. Y., & Chang, D. (2014). Semantically-based recommendation by using semantic clusters of users' viewing history (pp. 83–87). In *2014 International Conference on Big Data and Smart Computing, BIGCOMP 2014*. doi:10.1109/BIGCOMP.2014.6741412

Kushwaha, N., & Vyas, O. P. (2014). SemMovieRec: Extraction of Semantic Features of DBpedia for Recommender System (pp. 13:1–13:9). In *7th ACM India Computing Conference*. doi:10.1145/2675744.2675759

Landis, J. R., Koch, G. G. (1977), The measurement of observer agreement for categorical data. Biometrics, 33(1):159-174.

Lommatzsch, A., Kille, B., & Albayrak, S. (2013). Learning hybrid recommender models for heterogeneous semantic data. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13* (p. 275). New York, USA: ACM Press. doi:10.1145/2480362.2480420

Manoj Kumar, S., Anusha, K., & Santhi Sree, K. (2015). Semantic Web-based Recommendation: Experimental Results and Test Cases. *International Journal of Emerging Research in Management & Technology*, 4(6), (pp. 215–222).

Marie, N., Gandon, F., Legrand, D., & Ribi ere, M. (2013). Discovery Hub: A Discovery Engine on the Top of DBpedia (pp. 4:1–4:6). In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*, New York, NY, USA: ACM. doi:10.1145/2479787.2479820

Moreno, A., Ariza-Porras, C., Lago, P., Jiménez-Guarín, C. L., Castro, H., & Riveill, M. (2014). Hybrid model rating prediction with linked open data for recommender systems. *Communications in Computer and Information Science*, 475, (pp. 193–198). doi:10.1007/978-3-319-12024-9_26

Musto, C., Basile, P., Lops, P., De Gemmis, M., & Semeraro, G. (2014). Linked open data-enabled strategies for top-n recommendations (Vol. 1245, pp. 49–55). *Presented at the CEUR Workshop Proceedings.*

Nguyen, P., Tomeo, P., Di Noia, T., & Di Sciascio, E. (2015). An evaluation of SimRank and Personalized PageRank to build a recommender system for the Web of Data (pp. 1477–1482). In *Companion: Proceedings of the 24th International Conference on World Wide Web*. doi:10.1145/2740908.2742141

Ostuni, V. C., Di Noia, T., Di Sciascio, E., & Mirizzi, R. (2013). Top-N recommendations from implicit feedback leveraging linked open data. In *Proceedings of the 7th ACM conference on Recommender systems - RecSys '13* (pp. 85–92). New York, USA: ACM Press. doi:10.1145/2507157.2507172.

Ostuni, V. C., Di Noia, T., Mirizzi, R., Di Sciascio, E., & Noia, T. Di. (2014). A Linked Data Recommender System Using a Neighborhood-Based Graph Kernel. In *E-Commerce and Web Technologies SE-10*, 188, (pp. 89–100). doi:10.1007/978-3-319-10491-1_10

Passant, A. (2010). dbrec - Music Recommendations Using DBpedia. In *Lecture Notes in Computer Science Vol. 6497 LNCS*, (pp. 209–224). Springer. doi:10.1007/978-3-642-17749-1_14.

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook SE - 8* (pp. 1–2). Springer. doi:10.1007/978-0-387-85820-3_8

Ristoski, P., Mencía, E. L., & Paulheim, H. (2014). A hybrid multi-strategy recommender system using Linked Open Data. *Communications in Computer and Information Science*, 475, (pp. 150–156). doi:10.1007/978-3-319-12024-9_19

Schmachtenberg, M., Bizer, C., & Paulheim, H. (2014). Adoption of the Linked Data Best Practices in Different Topical Domains. In P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, D. Vrandečić, C. Goble (Eds.), *The Semantic Web – ISWC 2014 SE - 16* (Vol. 8796, pp. 245–260). Springer. doi:10.1007/978-3-319-11964-9_16.

Seco, N., Veale, T., & Hayes, J. (2004). An Intrinsic Information Content Metric for Semantic Similarity in WordNet. In *European Conference on Artificial Intelligence*. (pp. 1089–1090).

Schmachtenberg, M., Strufe, T., & Paulheim, H. (2014). Enhancing a Location-based Recommendation System by Enrichment with Structured Data from the Web. Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14) - WIMS '14, 1–12. https://doi.org/10.1145/2611040.2611080

Shani, G., & Gunawardana, A. (2011). Evaluating Recommendation Systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook SE - 8* (pp. 257–297). Springer. doi:10.1007/978-0-387-85820-3_8

Stankovic, M., Breitfuss, W., & Laublet, P. (2011). Discovering Relevant Topics Using DBPedia: Providing Non-obvious Recommendations. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology* (pp. 219–222). IEEE. doi:10.1109/WI-IAT.2011.32

Strobin, L., & Niewiadomski, A. (2014). Recommendations and object discovery in graph databases using path semantic analysis. In *Lecture Notes in Computer*. (Vol. 8467, pp. 793–804). Springer. doi:10.1007/978-3-319-07173-2_68

Vagliano, I., Figueroa, C., Rodriguez, O., Torchiano, M., Faron-Zucker, C., & Morisio, M. (2016). ReDyAl: A Dynamic Recommendation Algorithm based on Linked Data. In *3rd Workshop on New Trends in Content-based Recommender Systems - CBRecSys 2016* (pp. 31–39). Boston, MA, USA: CEUR Workshop Proceedings.

ENDNOTES

[1] http://natasha.polito.it/AlliedWI
[2] http://www.w3.org/2004/02/skos/
[3] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/
[4] https://wordnet.princeton.edu/
[5] http://www.w3.org/2004/02/skos/
[6] http://wiki.dbpedia.org/Datasets#h434-7
[7] http://natasha.polito.it/RSEvaluation/
[8] http://www.imdb.com/chart/top
[9] http://natasha.polito.it/RSEvaluation/faces/resultsdownload.xhtml