UNIVERSIDAD DEL CAUCA

SCUOLA DI DOTTORATO

PhD Course in Telematics Engineering

PhD Dissertation

Recommender Systems based on Linked Data

Cristhian Nicolás Figueroa Martínez

Coordinatore del corso di dottorato
prof. Alvaro Rendón Gallón

Tutors
prof. Juan Carlos Corrales - Universidad del Cauca. prof. Maurizio Morisio - Politecnico di Torino

December 2016

POLITECNICO DI TORINO

SCUOLA DI DOTTORATO

PhD Course in Ingegneria informatica e dei sistemi – XXVI ciclo

PhD Dissertation

Recommender Systems based on Linked Data

Cristhian Nicolás Figueroa Martínez

Coordinatore del corso di dottorato
prof. Matteo Sonza Reorda

Tutors
prof. Juan Carlos Corrales - Universidad del Cauca. prof. Maurizio Morisio - Politecnico di Torino

December 2016

To God, for being my guide, hope and light in my path.

To my parents because they fought by my side even when adversity seemed to give no respite.

To my siblings and my nephew for being my source of motivation and inspiration.

To all of my family because they trusted me and raised their prayers.

To my teachers who guided me in this educational process.

To my friends whom supported me and were a source of hope.

Acknowledgements

My deepest gratitude to my parents Efrén and Lidia, to my siblings Camilo, Carolina and Dayana, to my nephew José David, to my sister-in-love Katherine and all of my family for being by my side encouraging me and supporting me with all their patience and love. To my grandfather Carlos who trusted me because he was sure that I would reach my professional goals.

To my supervisors Prof. Maurizio Morisio and Prof. Juan Carlos Corrales for guiding me throughout my doctoral process and teaching me the basis of the scientific research.

To Prof. Álvaro Rendón Gallón for his kindness and availability to solve my concerns and situations during my master's and doctoral period at the Universidad del Cauca.

To professors Marco Torchiano and Maurizio Morisio for the great support and scientific collaboration during my doctoral stay at the Politecnico di Torino. Their support was fundamental in the successful completion of my doctoral studies.

To Henry Françoise Tarlin because of his management within the international relations office at the University of Cauca made possible the co-direction thesis agreement with the Politecnico di Torino.

To my friends and study fellows Hugo, Armando, Wilmar, Jaime, Carlos Mario, Oscar, Iacopo, Luca, Federico, Andrea, Syed, Rifat and Sham who gave me their friendship and support in several scientific projects and publications.

To the Joint Open Lab of Telecom Italia for giving me the opportunity to collaborate with them by applying some results of my thesis within their research projects.

To the University of Cauca and the Politecnico di Torino for giving me the opportunity to do this thesis in co-direction and obtain my doctoral degree in both institutions.

To COLCIENCIAS for the financing of my doctoral studies through the 511 scholarship.

Finally, I also wish to thank all of my friends and all the people who in one way or another contributed to make this doctoral dream possible.

Summary

Backgrounds: The increase in the amount of structured data published using the principles of Linked Data, means that now it is more likely to find resources in the Web of Data that describe real life concepts. However, discovering resources related to any given resource is still an open research area. This thesis studies Recommender Systems (RS) that use Linked Data as a source for generating recommendations exploiting the large amount of available resources and the relationships among them.

Aims: The main objective of this study was to propose a recommendation technique for resources considering semantic relationships between concepts from Linked Data. The specific objectives were: (*i*) Define semantic relationships derived from resources taking into account the knowledge found in Linked Data datasets. (*ii*) Determine semantic similarity measures based on the semantic relationships derived from resources. (*iii*) Propose an algorithm to dynamically generate automatic rankings of resources according to defined similarity measures.

Methodology: It was based on the recommendations of the Project management Institute and the Integral Model for Engineering Professionals (Universidad del Cauca). The first one for managing the project, and the second one for developing the experimental prototype. Accordingly, the main phases were: (*i*) Conceptual base generation for identifying the main problems, objectives and the project scope. A Systematic Literature Review was conducted for this phase, which highlighted the relationships and similarity measures among resources in Linked Data, and the main issues, features, and types of RS based on Linked Data. (*ii*) Solution development is about designing and developing the experimental prototype for testing the algorithms studied in this thesis.

Results: The main results obtained were: (*i*) The first Systematic Literature Review on RS based on Linked Data. (*ii*) A framework to execute and analyze recommendation algorithms based on Linked Data. (*iii*) A dynamic algorithm for resource recommendation based on on the knowledge of Linked Data relationships. (*iv*) A comparative study of algorithms for RS based on

Linked Data. (*v*) Two implementations of the proposed framework. One with graph-based algorithms and other with machine learning algorithms. (*vi*) The application of the framework to various scenarios to demonstrate its feasibility within the context of real applications.

Conclusions: (*i*) The proposed framework demonstrated to be useful for developing and evaluating different configurations of algorithms to create novel RS based on Linked Data suitable to users' requirements, applications, domains and contexts. (*ii*) The layered architecture of the proposed framework is also useful towards the reproducibility of the results for the research community. (*iii*) Linked data based RS are useful to present explanations of the recommendations, because of the graph structure of the datasets. (*iv*) Graph-based algorithms take advantage of intrinsic relationships among resources from Linked Data. Nevertheless, their execution time is still an open issue. Machine Learning algorithms are also suitable, they provide functions useful to deal with large amounts of data, so they can help to improve the performance (execution time) of the RS. However most of them need a training phase that require to know a priory the application domain in order to obtain reliable results. (*v*) A logical evolution of RS based on Linked Data is the combination of graph-based with machine learning algorithms to obtain accurate results while keeping low execution times. However, research and experimentation is still needed to explore more techniques from the vast amount of machine learning algorithms to determine the most suitable ones to deal with Linked Data.

Resumen

Antecedentes: El incremento en la cantidad de datos estructurados, que se encuentran publicados bajo los principios de los datos enlazados (Linked Data), demuestra que ahora es más fácil encontrar recursos que describan conceptos de la vida real en la Web de los datos. Sin embargo, descubrir recursos relacionados con un recurso determinado es aún un área abierta de investigación. Esta tesis, estudia los sistemas de recomendación (RS) que utilizan los datos enlazados como fuente para generar recomendaciones explotando la gran cantidad de recursos disponibles y las relaciones entre ellos.

Objetivos: El objetivo principal de este estudio fue proponer una técnica de recomendación que tenga en cuenta las relaciones semánticas entre conceptos de los datos enlazados (Linked Data). Los objetivos específicos fueron: (*i*) Definir relaciones semánticas derivadas de los recursos teniendo en cuenta el conocimiento encontrado en los conjuntos de datos de Linked Data. (*ii*) Determinar las medidas de similitud semánticas derivadas de esos recursos. (*iii*) Proponer un algoritmo para generar dinamicamente y automaticamente rankings de recursos de acuerdo con las relaciones de similitud definidas.

Metodología: la metodología estuvo orientada por las recomendaciones del PMI (Project Management Institute) y el Modelo Integral para un Profesional en Ingeniería de la Universidad del Cauca. El primero para gestionar el proyecto, y el segundo para desarrollar el prototipo experimental. De esta manera las principales fases fueron: (*i*) Generación de la base conceptual para identificar los problemas principales, objetivos, y los alcances del proyectos. Con este fin, una revisión sistemática de la literatura fue realizada, la cual permitió determinar as relaciones y medidas de similitud entre recursos de Linked Data, así como los principales problemas, características y tipos de RS basados en los datos enlazados. (*ii*) Desarrollo de la solución en la cual fue diseñado y desarrollado el prototipo experimental para probar los algoritmos estudiados en esta tesis.

Resultados: Los principales resultados fueron: (*i*) La primera revisión sistemática acerca de RS basados en los datos enlazados. (*ii*) Un entorno para ejecutar y

analizar algoritmos de recomendación basados en los datos enlazados. (*iii*) Un algoritmo dinámico para la recomendación de recursos basada en el conocimiento de las relaciones entre datos enlazados. (*iv*) Un estudio comparativo de los algoritmos para RS basados en los datos enlazados. (*v*) Dos implementaciones del entorno propuesto. Una con algoritmos basados en grafos y la otra con algoritmos de aprendizaje supervisado. (*vi*) La aplicación del entorno a varios escenarios para demostrar su factibilidad dentro del contexto de aplicaciones reales.

Conclusiones: (*i*) El entorno propuesto demostró su utilidad para desarrollar y evaluar diferentes configuraciones de algoritmos para crear RS novedosos basados en los datos enlazados adaptados a los requerimientos de los usuarios, aplicaciones, dominios y contextos. (*ii*) La arquitectura en capas del entorno propuesto es también útil para permitir que los resultados puedan ser reproducibles para la comunidad científica. (*iii*) Los RS basados en los datos enlazados son útiles para presentar explicaciones de las recomendaciones debido a la estructura de grafo que tienen los conjuntos de datos. (*iv*) Los algoritmos basados en grafos toman ventaja de las relaciones intrínsecas entre recursos de los datos enlazados. No obstante sus tiempos de ejecución son aún tema de investigación. Los algoritmos de aprendizaje supervisado también son adecuados, ellos proveen funciones útiles para tratar con grandes cantidades de datos, por lo tanto pueden ayudar a mejorar el rendimiento (tiempo de ejecución) de los RS. Sin embargo, ellos necesitan una fase de entrenamiento que requiere conocer a priori el dominio de aplicación para obtener resultados confiables. (*v*) Una evolución lógica de los RS basados en LD es la combinación de algoritmos basados en grafos y los de aprendizaje supervisado para obtener resultados confiables mientras mantienen bajos tiempos de ejecución. Sin embargo, aún es necesario llevar a cabo experimentación e investigación para explorar más técnicas de la gran cantidad de algoritmos de aprendizaje supervisado y determinar los más aptos para tratar con los datos enlazados aplicados a la recomendación de recursos.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Nowadays, RS are increasingly common in many application domains, as they use analytic technologies to suggest different items or topics that can be interesting to an end user. However, one of the biggest challenges in these systems is to generate recommendations from the large amount of heterogeneous data that can be extracted from the items. Accordingly, some RS have evolved to exploit the knowledge associated to the relationships between data of items and data obtained from different existing sources [2]. This evolution has been possible thanks to the rise of the Web supported by a set of best practices for publishing and connecting structured data on the Web known as *Linked Data* [3].

Linked Data principles have lead to semantically interlink and connect different resources at data level regardless the structure, authoring, location etc. Data published on the Web using Linked Data has resulted in a global data space called the Web of Data. Moreover, thanks to the efforts of the scientific community and the W3C Linked Open Data (LOD) project[1], more and more data have been published on the Web of Data, helping its growth and evolution.

This thesis studied RS that use Linked Data as a source for generating recommendations exploiting the large amount of available resources and the relationships between them.

First, a comprehensive state of the art is presented in order to identify and study frameworks and algorithms for RS that rely on Linked Data.

Second a framework named *AlLied* that makes available implementations of the most used algorithms for resource recommendation based on Linked Data is described. This framework is intended to use and test the recommendation algorithms in various domains and contexts, and to analyze their behavior under different conditions. Accordingly, the framework is suitable to compare the results of these algorithms both in performance and relevance, and to enable the development of

[1] `http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`

innovative applications on top of it.

Third, two implementations of the *AlLied* framework are described, including algorithms for generating candidate resources as well as for ranking and grouping them. The implemented algorithms into the *AlLied* framework are classified in graph-based algorithms, directly related with the graph structure of the Linked Data datasets, and machine learning algorithms which are capable of learning similarities between resources based on their relationships extracted from Linked Data.

Fourth, a new dynamic algorithm named *ReDyAl* for resource recommendation based on Linked Data is proposed. This algorithm considers the different relationships between resources and is able to choose the best strategy to find candidate resources to be recommended based on the implicit knowledge spread across the Linked Data relationships.

Furthermore, an experimentation was conducted to evaluate the accuracy and performance[2] of the algorithms for both implementations: graph-based and machine learning. Finally, this thesis presents some real use cases where the framework or part of it was tested.

1.1 Contributions

The main contributions of this thesis are:

- The first Systematic Literature Review on Recommender Systems based on Linked Data. Published in the Journal of Concurrency and Computation: Practice and Experience [4].

- A framework to execute and analyze recommendation algorithms based on Linked Data. Presented in a paper entitled "*Allied* A Framework for Executing Linked Data-based Recommendation Algorithms", accepted for publication on the 13 (3) 2017 issue of the International Journal on Semantic Web and Information Systems (IJSWIS) [5].

- A dynamic algorithm named *ReDyAl* for resource recommendation based on on the knowledge of Linked Data relationships. Published in a paper entitled "ReDyAl: A Dynamic Recommendation Algorithm based on Linked Data" in the 3rd Workshop on New Trends in Content-based Recommender Systems - CBRecSys 2016, within the most important conference about recommender systems RecSys 2016 [6].

[2]in this thesis performance is referred as the computational complexity in terms of execution time

- A set of datasets named LOD Matrixes useful for executing and testing machine learning algorithms in RS based on Linked Data.
- An implementation of graph-based algorithms for the *AlLied* framework.
- An implementation of machine learning algorithms for the *AlLied* framework.
- A grouping algorithm to categorize the recommendations by arranging the candidate resources into meaningful groups or contexts.
- A comparative study of algorithms for recommender systems based on Linked Data.
- The application of the framework to various scenarios to demonstrate its feasibility within the context of real applications.

1.2 Context

1.2.1 Problem Definition

Recommender Systems (RS) are software tools and techniques to make suggestions of items or objects to end users [2]. These items can be of different classes such as films, music tracks, news, messages on social networks, people, web resources (program applications or web services) among others. The most popular techniques for RS are: content-based, collaborative filtering, knowledge-based, and hybrid.

Content-based make suggestions taking into account the ratings that users give to items according to their preferences and considering also the content of these items (e.g. keywords, title, pixels, disk space, etc) [7]. Collaborative-filtering (CF) generate recommendations of items to a user taking into account ratings that users with similar preferences have given to the same set of items [8]. Knowledge-based infer and analyze similarities between user requirements and features of items described in a knowledge base. The knowledge base is useful to model users and items according to a specific application domain [9]. Hybrid RS combine one or more of the aforementioned techniques, aiming to address the problems that these techniques contain when they work separately. For example, CF methods suffer from the problem of the "new user" where new users with no ratings or with only a small number of ratings is probable. This, however, is not a limitation for content based methods since the prediction of the new items is focused on the description of their characteristics which generally is available [2]. Therefore by combining both techniques into a hybrid RS it is possible to mitigate their individual problems.

According to the work of Dell'Aglio et al., [9], knowledge-based RS have some advantages over other types of RS such as: 1) do not require lots of information about the user profiles to generate recommendations; 2) they do not suffer from the

problem known as cold start (when a new user or item is added to the system and does not contain enough information about previous ratings); and 3) they offer the possibility of showing "explanations" about the recommendations (i.e., the reason because a recommendation was generated). The main problems of the knowledge-based RS are the computational complexity due to the high cost of processing large amounts of data, and the high costs of construction, modeling, and maintenance of the knowledge base. Furthermore, the knowledge base depends on the application domain and may require frequent updates.

Consequently, a new kind of knowledge-based RS has emerged known thanks to the evolution of the Web towards a Web of data where different kinds of relationships can be established between resources. This new type of RS suggest items taking into account the knowledge of datasets published on the Web of data [10]. Web of data is a relatively new worldwide effort to create a web exposing and interlinking data that previously were isolated. The set of rules to create the Web of data are known as "Linked Data principles" [11]. Hence the name given to this type of knowledge-based RS is "Linked Data based RS".

Unlike traditional knowledge-based RS, the Linked Data based RS use datasets build, modeled, and maintained by different organizations and communities around the world. These datasets may contain knowledge form different domains and sources, and may be published on the web of data under the Linked Data principles.

However, until now the research works studied in this thesis (see Chapter 2 about the state of the art) still have some problems to generate recommendations with an acceptable level of accuracy for end users. For example, some Linked Data based RS still require information from both user profiles and descriptions of the items; others require knowledge bases to be frequently updated and maintained; others have high computational complexity; and others need a manual extraction of a subset of the knowledge bases representing a portion specialized on a specific domain of interest. Consequently, more research on how to apply the different techniques of RS and the web of data in real-world situations is required [12]. Hence, the main research question addressed in this thesis is:

How to recommend resources dynamically considering the knowledge of the web of data, analyzing their relationships and considering or not the application domain?

1.2.2 Motivating Scenario

Assume a scenario where a user of the RS is a developer of software applications. The developer requires to obtain heterogeneous web resources from different sources and application domains in order to be integrated into a new application. However, the main challenge for the developer is to find these resources that best suit with regard to his/she needs and the specific domain of the application that he/she is

developing. For example, in an application for the films domain the developer would prefer content related with actors, films, directors, writers, among others.

Therefore, the developer requires a system that supports the heterogeneity of data and recommendations of resources grouped according to different domains of applications, so that the developer can select and use them according to his/her convenience. In this case the user requirements may be expressed in a very abstract way, for instance indicating few parameters that the developer requires at certain time of the application development.

1.2.3 Scope

The study presented in this thesis is limited to knowledge-based RS that use Linked Data as source of knowledge to generate recommendations about resources. It does not consider those RS that require the user profile including historical view of items or user ratings. This selection is because the approaches considered in this thesis are intended as a solution for the cold start problem.

1.2.4 Thesis Structure

This thesis is structured as follows: Chapter 2 presents a short description of the conceptual foundation of the Web of Data and Recommender Systems (RS); the results of a systematic literature review conducted to identify the main research studies in this topic, and summarizes the current gaps found in the studies selected. Chapter 3 describes the conceptual architecture of the proposed framework named *AlLied* for recommendation based on Linked Data; Chapter 4 describes an implementation of the *AlLied* using graph-based algorithms and proposes a new dynamic algorithm, for resource recommendation based on Linked Data, named *ReDyAl*; Chapeter 5 describes the creation of a Linked Data based dataset and its implementation and configuration of machine learning algorithms within *AlLied*. Chapter 6 outlines the evaluation methodology and the results obtained in order to compare both implementations. Finally, Chapter 7 presents the main contributions, conclusions, application scenarios and future works of this study.

1.3 Summary

Linked Data principles have lead to semantically interlink and connect different resources at data level regardless the structure, authoring, location etc. Data published on the Web using Linked Data has resulted in a global data space called the Web of Data. This thesis studies RS that use Linked Data as a source for generating recommendations exploiting the large amount of available resources and the

relationships between them. This section presents the introduction, states the research problem, and a motivating scenario where Linked Data may be used to solve a problem for developers creating RS.

Chapter 2

State of the art

This chapter presents a Systematic Literature Review(SLR), which is one of the contributions of this thesis. The SLR was described in a paper entitled "A Systematic Literature Review of Linked Data-based Recommender Systems"[4] that has been published on the Journal of "Concurrency and Computations: Practice and Experience". A SLR is a form of secondary study that uses a well-defined methodology to identify, analyze and interpret all available evidence related to specific research questions in a way that is unbiased and (to a degree) repeatable [13, 14]. The chapter starts with a short description of the conceptual foundation of the Web of Data and Recommender Systems (RS); then it presents the results of the SLR conducted to identify the main research studies in this topic, and finally it summarizes the current gaps found in the studies selected.

2.1 Conceptual Foundation

This section presents the main conceptual foundations of the Web of Data and Recommender Systems (RS). An extended version, which presents a comprehensive overview of the technologies, standards, and principles of the Web of Data, as well as the classification and problems of RS is presented in Appendix A.

2.1.1 The Web of Data

The *Web of Data* is a subset of the World Wide Web based on the integration of a subset of the Semantic Web technology stack with existing standards of the World Wide Web [15]. Unlike the World Wide Web that consists of human-readable documents linked via hyperlinks, the *"Web of Data"* refers to a global space of structured and machine-readable data[16]. The Web of Data offers different types of links to give a meaning to each relationship between data, in this way the web is

taken to a semantic level where the knowledge about the relationships between data acquires value.

Linked Data

In 1994, Tim Berners-Lee[1] uncovered the need of introducing semantics into the Web to extend its capabilities and to publish structured data on it, which became known as *Semantic Web*. The set of good practices or principles for publishing and linking structured data on the Web is known as Linked Data. While the Semantic Web is the goal, Linked Data provides the means to make it reality [3]. The set of Linked Data principles are:

- Use URIs (Uniform Resource Identifier) as names for things.
- Use HTTP URIs, so that people can look up those names.
- Use of standard mechanisms to provide useful information when someone looks up a URI, for example *RDF* (Resource Description Framework) to represent data as graphs and *SPARQL* (SPARQL Protocol and RDF Query Language) to query Linked Data.
- Include links to other URIs, so that they can discover more things.

URIs are a fundamental concept for the Web architecture, intended to increase the value of the World Wide Web trough a "single global identification system" [17]. As constraint URIs should be unique so distinct resources must be assigned distinct URIs.

Resources are objects or concepts identified with a URI. To represent these resources there are various languages, but the most widely used is the RDF language. In this thesis the terms "concept" and "resource" are used indifferently to denote abstract "things" or objects of the real world.

Resource Description Framework

RDF is a recommendation of the W3C that provides a generic graph-based data model for describing resources, including their relationships with other resources [3].

The graph data model of the RDF framework is composed of triples or statements. Each triple contains a subject, a predicate, and an object. Triples assert facts about the resources [1]. In a triple the subject is an input resource from which an arc leaves, the predicate is a property (link) that labels the arc, and the object

[1] `http://www.w3.org/Talks/WWW94Tim`

is an output resource or literal (where the arc ends). Literals are resources that can be used for values such as strings, numbers and dates. RDF data can be written in different ways known as serialization e.g. RDF/XML, Notation-3 (N3), Turtle, N-Triples, RDFa, and RDF/JSON [3].

Linked Data datasets

A dataset can be seen as a database storing a collection of triples that may or not belong to a specific domain. More formally, Passant [18] has defined a dataset as: *"A dataset following the Linked Data principles is a graph G such as $G = (R, L, I)$ in which $R = \{r_1, r_2, \ldots, r_n\}$ is a set of resources –identified by their URI–, $L = \{l_1, l_2, \ldots, l_n\}$ is a set of typed links –identified by their URI– and $I = \{i_1, i_2, \ldots, i_n\}$ is a set of instances of these links between resources, such as $i_i = \langle l_j, r_a, r_b \rangle$"*

This definition assumes the interlinked structure of the data in a Linked Data dataset which is not only limited to interlink resources within a dataset, but also made possible to interlink datasets. In this way Linked Data has made possible to create ecosystems of datasets composed of interlinked structured data. Some examples of the most important datasets are: DBpedia, GeoNames, FOAF Profiles, MusicBrainz, WordNet, and DBPLP bibliography.

Linked Data endpoints

Endpoints are the mechanism used in Linked Data to provide access to the available datasets. Endpoints may be seen as interfaces to execute queries to the datasets in a similar way as in a database. The language to express query across diverse datasets is SPARQL which is the de facto language for interaction with Linked Data [19]. SPARQL is a language that contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions [20].

2.1.2 Recommender Systems

Recommender Systems (RS) are software tools that use analytic technologies to suggest different items of interest to an end user. These items can belong to different categories or types, e.g. songs, places, news, books, films, events, etc. According to Adomavicius and Tuzhilin [21]. Nowadays, RS are focused on the recommendation problem, which looks for guiding users in a personalized way to interesting items in a large space of possible options [7]. Typically RS are classified as: content-based, collaborative filtering, knowledge-based, and hybrid [2].

Classification of Recommender Systems

Content-based (CB) RS make suggestions taking into account the ratings that users give to items according to their preferences and considering also the content of these

items (e.g. keywords, title, pixels, disk space, etc) [7].

Collaborative-filtering (CF) RS are the recommenders most mature and widely adopted due to their good results and their easy implementation [22]. CF algorithms generates recommendations of items to a user taking into account ratings that users with similar preferences have given to these items [8].

Knowledge-based RS infer and analyze similarities between user requirements and features of items described in a knowledge base that models users and items according to a specific application domain [9]. Afterwards, the knowledge base is used to apply inference techniques to find similarities between user needs and items features.

Hybrid RS combine one or more of the aforementioned techniques in order to circumvent limitations of individual techniques. According to Felfernig et al., [8] these combinations may be performed in the following ways: implementing two types of RS separately and combine their results; adding features form KB recommenders to CF; and developing a unique RS integrating both techniques relying on probabilistic and statistical tools.

The main problems detected in these types of RS are summarized in table 2.1.

Approach	Problems
Content-based	- Limited analysis of the content: CB recommenders require a consistent description for each feature of the contents in order to match them with users' preferences. - Novelty: items should be previously rated, for this reason CB recommenders are not able to perform unexpected recommendations.
Collaborativ Filtering	- Scarcity: when an item contains low number of ratings with regard to the total number of existing items. - New user problem: this problem is also known as cold-start problem, where new users with no ratings or with only a small number of ratings is probable. - Privacy issues: users may distrust in RS that can be invasive in their profiles
Knowledge-based	- High costs for modeling, constructing and maintaining the knowledge that are used by RS. These knowledge bases may depend on the application domain that can frequently change requiring constant updates.
Hybrid	- Although, hybrid RS are able to produce more accurate results than CB and CF, hybrid RS have a poor performance.

Table 2.1. Summary of the main problems of RS

As shown in Table 2.1 traditional RS still have some problems that prevent them to generate accurate recommendations. For this reason in the last years a new kind

of RS named RS based on Linked Data have raised, which exploits the knowledge found on the Web of data extracting concepts stored in datasets and relating them in some way with items or objects to be recommended.

2.1.3 Recommender Systems and Linked Data

With the evolution of the Web towards a global space of connected and structured data, a new kind of knowledge-based RS has emerged known as Linked Data-based RS. This kind of RS suggest items taking into account the knowledge of datasets published under the Linked Data principles. The main benefit of using Linked Data as a source for generating recommendations is the large amount of available concepts and their links that can be used to infer relationships more effectively in comparison to derive the same kind of relationships from text [10].

As Linked Data information is machine readable it is possible to query datasets on a fine-grained level in order to collect information without having to take manual actions, therefore information is explicitly represented. This allows recommender systems to apply reasoning techniques when querying datasets and make implicit knowledge explicit. A complete state of the art of Linked data based RS is presented in chapter 2, which classify the current approaches for recommendation based on Linked Data as well as the algorithms they use.

2.2 Systematic Literature Review

The SLR described in this chapter is, unlike other works reporting the state of the art in RS [21, 23, 7, 24], the first to study RS that obtain information from Linked Data in order to generate recommendations. Consequently, this chapter summarizes the state of the art in RS that use structured data published as Linked Data for providing recommendations of items from diverse domains. An extended version of this SLR may be found in the research paper [4]. The SLR considers the most relevant research problems addressed and classifies RS according to how Linked Data has been used to provide recommendations. Furthermore, it analyzes contributions, limitations, application domains, evaluation techniques, and directions proposed for future research. There are still many open challenges with regard to RS based on Linked Data in order to be efficient for real applications. The main ones are personalization of recommendations; use of more datasets considering the heterogeneity introduced; creation of new hybrid RS for adding information; definition of more advanced similarity measures that take into account the large amount of data in Linked Data datasets; and implementation of testbeds to study evaluation techniques and to assess the accuracy scalability and computational complexity of RS.

2.2.1 Research Methodology

The methodology followed in this thesis to study the state of the art is based on the guidelines set out by Kitchenham and Charters [14] for systematic literature reviews in software engineering. These guidelines provide a verifiable method of summarizing existing approaches as well as identifying challenges and future directions in the current research. Figure 2.1 presents the protocol for this systematic literature review.

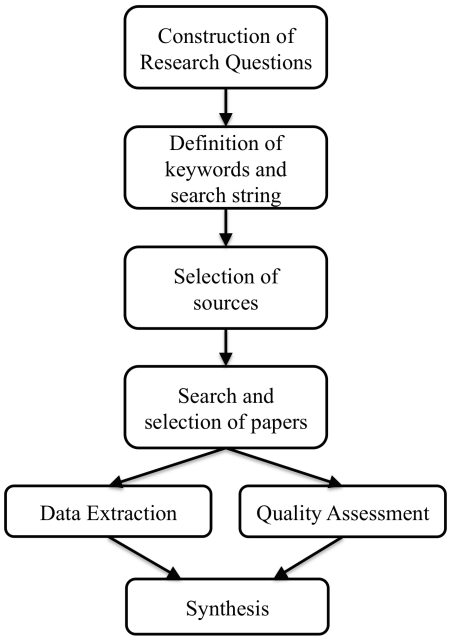


Figure 2.1. Systematic literature review at a glance

This protocol was defined in order to setup the steps to conduct the SLR. The goal of this SLR review is to understand how the implicit knowledge, stored in Linked Data datasets and represented as concepts and relations between them, can be exploited to make recommendations. Consequently, the following research questions have been defined:

RQ1 What studies present RS based on Linked Data?

RQ2 What challenges and problems have been faced by researchers in this area?

RQ3 What contributions have already been proposed (e.g. algorithms, frameworks, engines)?

RQ4 How is Linked Data used to provide recommendations?

RQ5 What algorithms have been used for RS based on Linked Data?

RQ6 What application domains have been considered?

RQ7 What criteria and techniques are used for evaluation?

RQ8 Which directions are the most promising for future research?

2.3 Results of the SLR

This section summarizes the relevant information found in the studies selected in order to answer the proposed research questions.

2.3.1 Included Studies

RQ1 regards the studies that present RS based on Linked Data. A set of 69 papers to include in the systematic literature review was retrieved, corresponding to 52 unique primary studies (a study is a unique research work that can include one or more papers). These studies were published in conferences, workshops and journals between 2004 and 2015. The final set of selected papers and corresponding studies, as well as the set of excluded papers are presented in a Appendix B.

2.3.2 Research Problems

RQ2 deals with research problems in the RS domain that researchers intended to solve by proposing approaches based on Linked Data. The lack of semantic information and its complexity were the most notorious problems in RS. Lack of semantics regards the need for rich semantic information about items. This is the main reason to devise novel strategies to represent items and user profiles using diverse semantic techniques exploiting several knowledge sources from the Linked Data cloud.

The complexity and heterogeneity of information and the subsequent cost of maintenance of knowledge bases makes Linked Data a suitable solution that uses publicly available knowledge bases that are continuously growing and maintained by third parties. However, this poses new challenges, for example the need for mechanisms to assure the reliability of these knowledge bases that are used to describe user profiles and items and to generate recommendations.

Domain dependency is another problem that has been also addressed by using Linked Data because it allows the possibility to exploit information from different datasets that can be domain-independent or belong to diverse domains. In fact this is one reason why the most used dataset is DBpedia as it is the most generic dataset that can be used for cross-domain RS. Nonetheless, some studies still report this problem as future work.

Computational complexity is a question that has not been widely addressed in the studies considered in this systematic literature review and remains as an open

issue because most of the studies have concentrated only on semantic enrichment of items and inclusion of Linked Data datasets. Computational complexity needs to be addressed more because in RS not only accuracy is important, but also scalability and responsiveness. For example, this problem can be critical in RS for mobile scenarios where users demand fast response times.

Other problems such as usability, cold-start, data quality and data sparsity have been addressed by combining with Linked Data various techniques based on natural language processing, reasoning or social network resources and creating hybrid RS that exploit both collaborative filtering and content-based approaches.

2.3.3 Contributions

RQ3 inquires about the contributions proposed in RS based on Linked Data. The analysis showed that the majority of studies are focused on providing new algorithms, but also on defining or extending a similarity measure of an ontology. Furthermore, adaptation, combination or extension to algorithms is quite often addressed together with information aggregation or enrichment. Accordingly, Linked Data can be used in RS for several purposes such as:

- Defining different similarity functions between items or users by exploiting the large data available in the Linked Data cloud and the vast relationships already established such as properties or context-based categories. In this way, it is possible to extract semantic information from textual descriptions or other textual properties about the items in order to find semantic similarities based on the information stored in interlinked vocabularies of Linked Data. This can be useful in RS based on collaborative filtering to improve the neighborhood formation in user-to-user or item-to-item.

- Generating serendipitous recommendations, for example to recommend items that are not part of the user's personal data cloud, i.e. suggest new, possibly unknown items, to the user; or to guide users in the process of the exploration of the search space giving the possibility for serendipitous discovery of unknown information (for exploratory search systems).

- Offering the explanation of the recommendations given to the users by following the linked-data paths among the recommended items. In this way, users can understand the relationship between the recommended items and why these items were recommended.

- Domain-independency when creating RS as it is possible to access data from Linked Data datasets from different domains.

- Enrichment of information sources such as databases, repositories, registries etc with information obtained from Linked Data datasets which manage huge amounts of, linked open data triples. It offers the possibility to enrich graphs representing users and/or items with new properties in order to improve graph-based recommendation algorithms. Additionally, it helps to mitigate the new-user, new-item and sparsity problems.
- Annotating items and users with information from multiple sources facilitate RS to suggest items from different sources without changing their inner recommendation algorithms. Using such a semantic-based knowledge representation, recommendation algorithms can be designed independently from the domain of discourse.
- Obtaining hierarchical representation of items because the topic distribution that some Linked Data datasets offer. In this way, RS can base their recommendation on the exploration of items belonging to similar categories.

2.3.4 Use of Linked Data

Another interesting aspect that was studied is the use of Linked Data in RS, as underlined by RQ4. The studies selected were classified according to the way they used Linked Data to produce recommendations and grouped them into:

Linked Data driven RS that rely on the knowledge of the Linked Data to provide recommendations. For example, RS that calculate a semantic similarity based on diverse relationships that can be found between concepts of Linked Data datasets and are related to features or descriptions of items. Such relationships can be paths, links or shared topics among a set of items. This category can also include RS that use other techniques applied on data obtained from Linked Data datasets, for example weight spreading activation, vector space model (VSM), SVM, LDA and random indexing.

Hybrid RS that exploit Linked Data to perform some operations that can be used or not used to provide recommendations. This means that Hybrid RS include Linked Data driven RS, which use recommendation techniques that rely on Linked Data, and RS that use Linked Data in other operations (not necessarily for recommending) that can be preliminary to the recommendation process (e.g. to aggregate more information from other datasets, to describe user profiles or to annotate raw data in order to extract information to be integrated and used for recommending).

Representation only RS in this category exploit the RDF format to represent data and use at least one vocabulary or ontology to express the underlying semantics. However, no information is extracted from other dataset and Linked

Data are not used to provide recommendations. An example is an RS that represents the information about the users according to FOAF vocabulary but does not exploits Linked Data for other operations.

Exploratory search These systems are not RS, but their main duty is to assist users to explore knowledge and to suggest relevant to a topic or concept. Exploratory search systems and RS use Linked Data in a very similar way, although the key difference is that exploratory search systems still require an explicit input query (commonly a set of keywords). Additionally, users in these systems are not only interested in finding items, but also in learning, discovering and understanding novel knowledge on complex or unknown topics [25].

Each study may be assigned to more than one category, i.e. it can be both Linked Data driven and hybrid, or both exploratory search and Linked Data driven. The only exception is for the representation only category, in which studies cannot belong to other categories.

Category	Number of studies
Linked Data driven	37
Hybrid	29
Hybrid and Linked Data driven	21
Linked Data driven only	13
Representation only	10
Hybrid only	6
Exploratory search	4
Exploratory search and Linked Data driven	4
Exploratory search only	0

Table 2.2. Distribution of studies according to the use of Linked Data

Table 2.2 shows that most of the studies considered are Linked Data driven, and roughly 60% of them are also hybrid. Only 20% of hybrid studies were hybrid only, while the rest are also Linked Data driven. Moreover, 10 studies are representation only and just 4 exploratory search systems were included in the systematic literature review. All of the exploratory search studies are also Linked Data driven. This finding is consistent with the focus of the systematic literature review, which is on RS using Linked Data. It is worth noting that exploratory search is a broader topic; this thesis only considers the exploratory systems that recommend concepts to users.

The two most interesting categories are Linked Data driven and hybrid. Figure 2.2 shows the different techniques used by the studies in the first category to provide recommendations. The majority of them rely on datasets or on a similarity measure

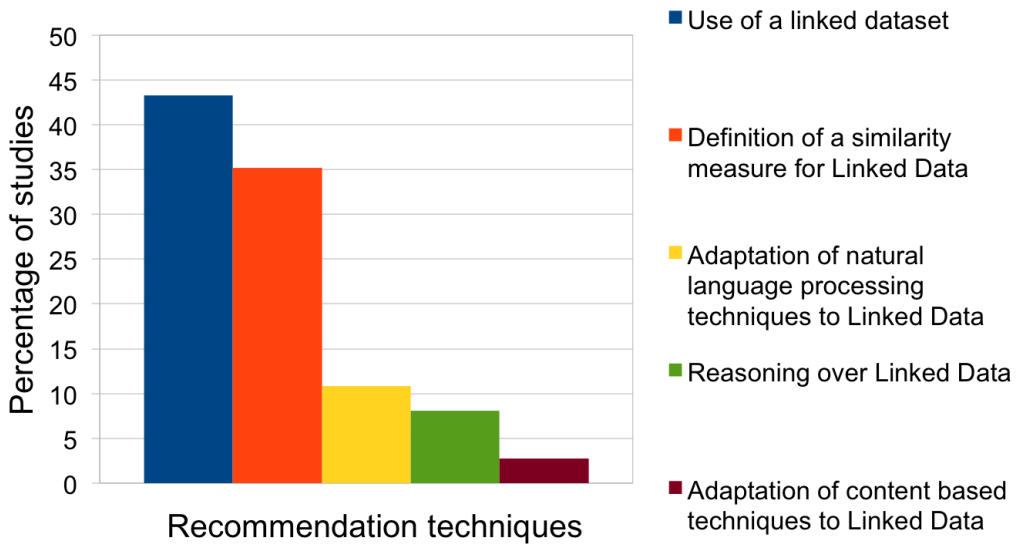


Figure 2.2. Distribution of Linked Data driven studies according to the recommendation techniques that they exploit (percentages refer to the total number of Linked Data driven studies)

(respectively about 43% and 35%), while the remaining 22% adapt natural language processing or content based techniques or exploit reasoning.

Table 2.3 describes each category including the most important studies that adopted these strategies, as well as their advantages and disadvantages. The numbers of the studies corresponds to the identifiers in the Appendix B.

Most of the studies belong to the first category, and many belong to both the first and the second category. These two categories are also the most interesting as they include RS to better exploit the advantages provided by Linked Data in order to reach best results. In this thesis techniques to provide recommendations relying on Linked Data were studied and slightly less than half of Linked Data driven RS used a dataset, almost one third define a similarity measure for Linked Data, while others adapt natural language processing or content based methods or use reasoning.

With reference to the techniques used together with Linked Data, it was found that natural language processing and collaborative filtering are the most used (both account for about one third of hybrid RS) as they intended to to provide personalized suggestions of items tailored to the preferences of individual users.

Other techniques are less common (less than 15%) and they are reasoning, use of social network resources and content based methods. Reasoning has not been

Approach	Techniques	Advantages	Disadvantages
Linked Data-driven	- *Graph based*: weight spreading activation (S17), semantic exploration in an RDF graph (S29, S10, S3, S9, S19), and projections (S23) - *Reasoning*: (S1, S51) - *Probabilistic*: Matrix item-user (S29, S35, S31, S13, S37, S10), Scaling methods (S29) and topic discovery (S2)	- Generating serendipitous recommendations - Offering explanations of the recommendations following the linked-data paths - Creating domain-independent RS - Exploiting hierarchical information about items to categorize recommendations	- High cost of exploiting semantic features due to inconsistency of LD datasets - No personalization - No contextual information - High computational complexity - Need for manual operation - Need for dataset customization to address the computational complexity
Hybrid	- *Collaborative Filtering and Linked Data*: (S2, S4, S12, S25, S27, S3, S28, S26, S30, S35) - *Information aggregation and Linked Data*: opinions (S16), ratings (S19), and social tags (S32) - *Probabilistic methods and Linked Data*: Random Indexing (S10), VSM (S47, S31, S35), LDA (S35), Implicit feedback (S25), SVM (S13), Structure-based statistical semantics (S37)	- Overcoming the data sparsity problem - Allowing collaborative filtering RS to address the cold start problem	- High computational complexity
Representati Only	- Item/user information representation using RDF-based ontologies $S36, S38, S20, S40, S14, S1$	- Improving scalability and reusability of ontologies - Easing data integration - Enabling complex queries	- Difficult to reuse the already available knowledge in the Linked Data Cloud
Explorative Search	- Set nodes and associated lists (S49, S39, S34) - Spreading activation to typed graphs and graph sampling technique (S11)	- Enabling self-explanation of the recommendations	- No automation of the recommendation because explorative search approaches require frequent interaction with the user

Table 2.3. Classification of Linked Data-based RS approaches

widely used as its quality is still insufficient and its coverage is not enough broad at the level of system components and knowledge elements [26]. Therefore one solution is to develop RS based on reasoning-oriented natural language processing enriched with multilingual sources and able to support knowledge sources generated largely by people as Linked Data datasets.

As for the datasets used in the studies selected, DBpedia was the most used Linked Data dataset. This is because DBpedia is a generic dataset and most of the studies are domain independent that need to be evaluated in diverse scenarios. DBpedia is one of the biggest datasets that is frequently updated as it obtains data from Wikipedia that continuously grows into one of the central knowledge sources [27]. It makes Dbpedia multimodal and suitable for RS that need to be domain independent and for knowledge based RS where complexity and cost of maintenance of the knowledge base is high. However for RS of a single domain it is better to use specific datasets but always implementing a linking interface with generic datasets in order to resolve ambiguities, or to exploit unknown semantic relationships.

2.3.5 Algorithms for RS based on Linked Data

In order to address RQ5, the selected studies were classified also according to the type of the algorithms they used. In this thesis the RS the selected studies were classified in five main types: graph-based, machine learning, memory-based, probabilistic, and others. Figure 2.3 shows the variety of algorithms for recommendation reported in the selected studies.

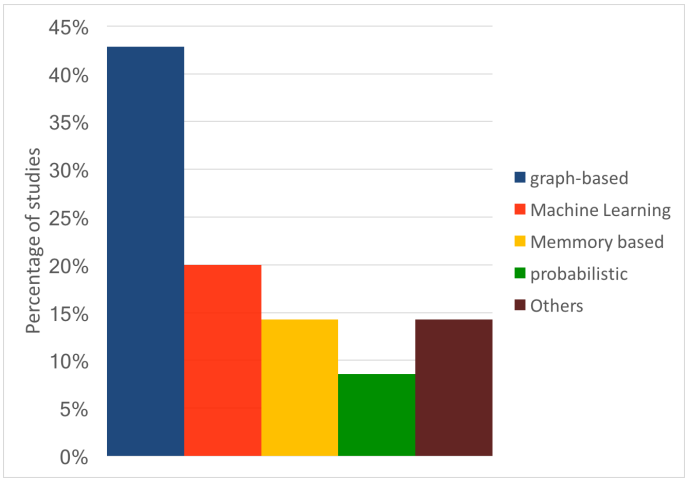


Figure 2.3. Distribution of studies according to the algorithms they used for recommendation

The two main types of algorithms are the graph-based and the machine learning algorithms, accounting for 42.9% and 20% of the studies respectively. Graph-based

algorithms were expected to be the most common type because of the graph structure of Linked Data. Machine learning algorithms are the second one because the data mining algorithms are the most used in general for any type of RS.

Graph-based: this is the most common type of algorithms used in RS based on Linked Data. These algorithms exploit the graph structure of Linked Data datasets for computing relevance scores for items represented as nodes in a graph. Algorithms in this category are classified on semantic exploration and path-based: (*i*) *Semantic Exploration:* explore the graph structure of LD datasets using structural relationships to compute distances and generate recommendations. For example, hyProximity, dbRec, pager rank, semantic clustering and the Vector Space Model (VSM). (*ii*) *Path-based:* use information about semantic paths within a RDF graph structure to compute similarities useful to produce recommendations. For example, spreading activation, random walk, and path-weights for vertex discovery.

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 in Supervised learning and unsupervised learning: (*i*) *Supervised:* a model is prepared through a training process where it produces predictions [28]. These algorithms predict class labels from attributes. For example, kNN, decision trees, logistic regression, support vector machine (SVM), random forest, naive Bayes, and bayesian classifiers. (*ii*) *Unsupervised:* unlike the supervised, input data is not labelled and does not have a known result, so the aim of these algorithms is to try to discover the structure or distribution of the data [28]. For example, K-Means, Fuzy-C-Means (FCM), self organizing map (SOM), and principal component analysis (PCA).

Memory-based: algorithms for rating predictions based on the entire collection of previously rated path queries. For example, rating prediction, singular value decomposition (SVD), impolitic feedback, and matrix factorization.

Probabilistic: recommendation algorithms based on probabilistic techniques applied to Linked Data such as Latent Dirichlet Allocation (LDA), Random Indexing (RI), bayesian ranking, and beta probability distribution.

Others: this group is composed of other types of algorithms that were found with less frequency in the SLR. For example, evolutionary computation, automated planning, semantic reasoning, and social network analysis (SNA). (*i*) *Evolutionary computation:* stochastic methods inspired from natural evolution such as genetic algorithm, biological classification and particle swarm optimization.

(*ii*) *Automated planning:* use artificial intelligence to create strategies that are executed by intelligent agents. (*iii*) *Semantic reasoning:* based on rules to infer logical consequences from a set of asserted facts or axioms. (*iv*) *Social network analysis:* exploit relationships found in social networks related with items and users.

Table 2.4 shows the classification of the algorithms with the most important studies as well as their advantages and disadvantages.

This literature review is limited to the most popular types of algorithms used in RS based on Linked Data: graph-based and machine learning.

2.3.6 Application Domains

RQ6 concerns the application domains considered by RS based on Linked Data so far. In this regard despite that 12 domains were identified, most of the RS are domain independent (slightly more than one fifth of the studies). This is because most of the recommendation algorithms proposed can be applied in diverse domains by only changing the dataset or taking only a portion of it in order to obtain the data to generate the recommendations.

However, items of music, tourism and movies are the most recommended as these belong to common domains in which there is a large amount of data and state-of-the-art datasets available, which allow the researchers to compare their results with several works developed in the community.

Accordingly, in a number of cases the domain impact also on datasets because they require a reduction of information, i.e, only a subset of concepts is considered, which requires offline processing and more effort to maintain the dataset even if it improves the performance. For example, Passant developed a RS named *dbrec* [29], which required to manually extract a subset of the data of DBpedia related with bands and musical artists.

2.3.7 Evaluation Techniques

RQ7 regards the evaluation techniques used to study RS based on Linked Data. In this thesis the evaluation techniques were classified into two types: accuracy and performance. Accuracy evaluates recommendations according to their relevance for final users, while performance measures the execution time required to produce them.

With regard to accuracy, the results demonstrate that researchers are more interested in evaluations made by final users than in comparisons with similar methods. This result was expected because usefulness of recommendations depends more on final user preferences than on comparing with similar approaches where evaluation may be biased as researchers must trust the results obtained. Therefore future

Type	Algorithm	Advantages	Disadvantages
Graph-based	- *Semantic exploration:* HyProximity [10], DbRec [29, 30, 31], page rank [32, 33, 34], semantic clustering [35], and VSM [32, 36, 37, 38, 39]. - *Path-based:* spreading activation [40, 41, 39, 42, 43, 44, 45, 46], random walk [47]; path-weights for vertex discovery [48].	- Serendipitous recommendations. - Explanations of recommendations following LD paths. - Creation of domain-independent RS. - Exploiting hierarchical information to categorize recommendations.	- High cost of exploiting semantic features due to inconsistency of LD datasets. - No contextual information. - High computational complexity for large datasets. - Need for dataset customization to address the computational complexity.
Machine Learning	- *Supervised*: kNN[49, 50], decision trees [51, 49, 34, 37], logistic regression [51, 52, 32, 53, 54], SVM [55, 56, 57, 37], random forest[51, 32], naive Bayes [58] and bayesian classifiers [59]. - *Unsupervised*: K-Means [52, 60], fuzzy-C means [57], SOM [57]; PCA[57].	- A large number of algorithms already developed to configure for recommendations. - Some algorithms can deal with big 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 LD to enrich data of items or users, so the intrinsic semantic structure of the LD is not taken into account.
Memory-based	Rating prediction[55, 52, 32]; SVD [52, 61, 62]; implicit feedback; and matrix factorization[63, 64]	- Well stablished algorithms for RS based mainly on CF approaches, e.g., [52, 65, 66, 30, 67, 68, 54, 69, 70]. - Easy to implement/use.	Cold-start problem for users or items. - Time-consuming algorithms
Probabilistic	LDA [71, 37, 54]; random indexing; bayesian ranking[72]; and beta probability distribution [73].	- Detect patterns within data for profiling and rating estimation.	- Cold-start problem for users or items
Others	- *Evolutionary computation:* genetic algorithms [37, 64], biological classification [70], automated planning [74] , semantic reasoning [75, 76, 77] and social network analysis [72, 59].	- Much of them use heuristics that may reduce the execution time based on optimization techniques.	- These algorithms have not been widely studied and used for RS based on LD knowledge. They need more research to evaluate their performance on RS based on LD.

Table 2.4. Classification of algorithms for Linked Data-based RS

methodologies of evaluation should be user-centered in order to assure the quality of the results of RS.

Additionally, as expected most of the studies selected were more likely to evaluate their recommendations applying traditional methods of information retrieval such as Precision and Recall that are focused on percentages of true positives, false negatives, and false positives.

Interestingly, few works evaluated the performance of RS, which is a critical factor specially for applications that need responses with short timeouts. Therefore it is still an open issue considering that accessing to Linked Data datasets in most cases is time consuming and requires that researchers download dumps of the datasets to access them in local repositories.

2.3.8 Future Works

RQ8 aimed to uncover the most promising directions for future research on RS based on Linked Data. The most frequently future works were the personalization of recommendations, the use of more datasets, and the creation of hybrid RS.

The lack of personalization of recommendations is still a common drawback in Linked Data-based RS. It concerns the fact that different users obtain the same set of results with the same input parameters. To solve this drawback some RS need explicit feed back from users in order to differentiate the results based on information about the user's profile (e.g. browsing history, favorite music genre, etc).

However these approaches force the user to perform extra work like rating items or building an exhaustive user profiles. Consequently, there is a need of non-invasive personalization approaches supported by Linked Data in order to obtain implicit information from the neighborhood relationships user-to-user, item-to-item and user-to-item. These relationships can be inferred from the links between concepts of Linked Data datasets related with properties of items and users.

Using more datasets is needed in order to increase the base of knowledge to produce recommendations. There are some limitations of the current Linked Data-based RS with regard to the use of Linked Data datasets such as: restricted access, poor reliability, computational complexity, low coverage of languages, domain dependency and the need for installing a local copy of the dataset. For this reason, it is important to investigate new ways to integrate different datasets in order to:

(*i*) extend the knowledge base allowing the RS to access to other datasets in case that the main dataset fails or the data are not reliable; (*ii*) create scalable RS because they can be adapted to other domains by only accessing to the appropriate dataset (*iii*) and improve the performance by selecting datasets with better response time.

The creation of hybrid RS is not a new proposal, as could be seen in Section

2.3.4, combining diverse techniques of recommendation with Linked Data-based approaches is a frequent practice in the studies selected. However, it is still an open issue because it is necessary to investigate which combinations of techniques are more suitable for a RS applied in diverse contexts. For example, combining Linked Data-based RS with social-based RS can be a good choose for applications that require information about the users and their inter-relationships. In this way, RS can access information that sometimes is not available in Linked Data datasets such as items rating information, user profiles, and other social information.

The inclusion of user profile information (user profiling) is another aspect that is not widely considered in Linked Data recommender systems. The idea behind the user profiling is to obtain a meaningful concept driven representation of user preferences in order to enable more precise specifications of user's preferences with less ambiguity. Therefore, this can be also useful to contribute to the personalization of Linked Data-based RS.

The automatic selection of the appropriate dataset according to the type of items or the application domain is another challenge that intend to improve the quality of recommendations. This dynamic process of selection can help the algorithms to choose the best strategy to find candidate items to be recommended based on the implicit knowledge contained in Linked Data and the relationships with properties of items and users.

As a consequence, it is also important to study new similarity measures and techniques able to automatically combine information from different datasets and to deal with the diversity of data in these datasets. Furthermore, it can be possible to create a statistical models of user interests to overcome the topical diversity of rated items.

Finally, there is still a need for building testbeds in order to allow for rigorous, transparent, and replicable testing and for studying new techniques (or adaptation of those existing) for evaluating the accuracy and computational complexity of RS based on Linked Data. This also must consider that Linked Data-based RS may access to large amounts of information and that links among items can be unknown to the users. Additionally, large-scale RS should be also evaluated in terms of the ability to scale

2.3.9 Current gaps

Despite the growing interest in RS based on LD, they still have some problems to generate recommendations with an acceptable level of accuracy to the users. Table 2.5 shows the main problems or gaps found in the studies selected grouped into the following main types: datasets, manual operations, algorithms, and computational complexity.

Approach	Problems
Datasets	Problems related to the sources of data used by RS: - Need to create a local copy of the full data source because the public data sources often provide limited results, restricted access and large response times [29, 56, 78, 65]. - Need for manual operations to review and correct data due to the low reliability of public datasets [18, 29, 79]. - Sometimes the RS can only access to a limited portion of the knowledge due to a restriction of the data space to only one dataset or to a specific application domain [80, 81, 82, 83].
Manual Operations	Problems related to manual operations that users need to perform in order to obtain recommendations: - Manual selection of relevant concepts of interest for a specific application domain. This task is difficult an tedious considering the large amount of data that a typical dataset in Linked Data may contain [10, 18, 29, 79, 81, 83, 84]. - Manual ranking of the results: some RS do not perform the ranking of their results, requiring the user to sort them [85]. - Need of user's feedback to generate the recommendations [9, 80, 81, 47, 31]
Algorithms	Problems related with the algorithms used by RS: - Graph-based algorithms for RS suffer from high computational complexity for exploiting semantic features due to the huge data and inconsistency of LD datasets [10, 29]. - Machine Learning algorithms are time-consuming for the training phase, additionally, some of them only use LD for representation only so the intrinsic semantic structure of LD is not taken into account [51, 49, 86]. - Other algorithms require user's profile information to produce recommendation, they suffer from the cold-start problem [55, 52, 63]. - Existing hybrid recommendation techniques are not organized in a conceptual architecture based on their functionalities, which would be useful to execute and test various configurations of algorithms for creating novel RS [87, 63, 52, 51].
Computatior Complexity	In this thesis the term computational complexity is referred to the long response times that RS based on Linked Data required due to the high computational demands to analyze large amounts of data related with the items or concepts to be recommended. Moreover, other factor that impact in the response time is the poor performance to the access points or endpoints of public datasets in Linked Data [10, 29, 31]

Table 2.5. Summary of the gaps of RS based on Linked Data

2.4 Summary

In this chapter the most relevant problems that RS intended to solve, the way in which studies addressed these problems using Linked Data, their contributions, application domains and evaluation techniques that they applied to assess their recommendations were considered. Analyzing these aspects, the current limitations and possible directions of future research were deducted.

Chapter 3

AlLied: A Framework for Executing Resource Recommendation Algorithms based on Linked Data

As stated in Chapter 2, many algorithms have been developed in recent years to recommend resources (related with web resources) based on Linked Data. This chapter presents the *AlLied* framework, which includes the implementation of known algorithms for resource recommendation based on Linked Data. These algorithms are integrated as plugins to execute specific tasks in the process of recommendation.

Accordingly, the framework is suitable to compare the results of different configurations of these algorithms, and to enable the development of innovative applications on top of it. In this way, the framework constitutes an environment to select, evaluate, and create algorithms to recommend resources belonging to different contexts and application domains that can be executed within the same context and with different configuration parameters. Furthermore, chapters 4 and 5 present two implementations of the *AlLied* framework: with graph-based algorithms and with machine learning algorithms.

3.1 Architecture of the *AlLied* framework

The design of the *AlLied* framework was based firstly on the main conceptual architectures for developing Semantic Web applications, and secondly on the main steps of the recommendation process identified on the systematic literature review conducted in this thesis.

- *Conceptual architectures for developing Semantic Web applications:* provide a conceptual foundation to create Semantic Web applications as well as other

architectures on the top of them. They were proposed because of the lack of simplifying the development and deployment of Semantic Web applications, which has been an obstacle for real-world adoption of Semantic Web technologies and the Web of Data [88]. Two of the most known conceptual architectures for the Semantic Web are the Semantic Web stack and the conceptual architecture for applications on the Web of Data.

The Semantic Web stack [89] is a layered model representing the architecture of the Semantic Web, which defines relationships among the technologies and languages essential for the Semantic Web. This model may be divided into three layers, a bottom layer that provides the base for the Semantic Web for writing structured data with user-defined vocabularies; a middle layer for the implementation of the Semantic Web core techniques such as ontology languages, and query languages; and a top layer that provides a user interface for applications as well as enhancements to the lower layers though proof validation and trusting operations.

The conceptual architecture for applications on the Web of Data [88] is a component-based, Conceptual architecture for Semantic Web applications, which describes the high-level components that implements the functionality that differentiates Semantic Web applications. This architecture contains of seven components: 1) graph access layer: the interface for the application logic to access local or remote data sources; 2) RDF store: the persistent storage or RDF and other graph-based data; 3) Data homogenization service: address the structural, syntactic and semantic heterogeneity of data; 4) data discovery service: implements automatic discovery and retrieval of external data; 5) graph query language: performs graph-based queries on the data in addition to search on unstructured data; 6)graph-based navigation interface: provides a human accessible interface to navigate the graph-based data; and 7)structured data authoring interface: useful to enter new data, edit existing data, and import or export data.

Both conceptual architectures may be divided into three common layers: 1) knowledge base management represented by the bottom layer in the first architecture, and by component 2 in the second architecture; 2) Recommender System Management, represented by the intermediate layer in the first architecture, and by components 1,3,4, and 5 of the second architecture; and 3) user interface and applications layer represented by the top layer of the first architecture and components 6 and 7 of the second architecture.

- *The recommendation process:* The recommendation process is the set of steps followed by most of the RS studied on the state of the art in order to produce recommendations. In this thesis, the recommendation process was generalized based on the common steps of the RS studied in the SLR presented in 2. This

recommendation process may be decomposed in a sequence of four steps as shown in Figure 3.1.

Figure 3.1. Steps of the recommendation process

These steps represent the tasks that a RS executes to produce recommendations. The first step is intended to generate a set of candidate resources CR that maintain semantic relationships with an initial resource ir. The initial resource may be any object or resource identified with a URI. The semantic relationships may be seen as direct or indirect links between two resources in a Linked Data dataset. The second step sorts the candidate resources generated in the previous step from highest to lowest 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. Up to this point the candidate resources are ranked but a ranked list is too general and does not provide a distinction between the results according to contexts or application domains. For this reason, the third step groups the ranked resources into meaningful clusters or contexts based on hierarchical relationships inferred from the Linked Data. This step may be also swapped with the ranking step, i.e. firstly grouping the candidate resources, and then ranking the candidate resources for each group separately. Finally, the last step presents the results through different interfaces that allow the end-users to visualize the recommendations.

The architecture *AlLied* framework was designed taking into account the conceptual architectures for Semantic Web applications, and the main steps of the recommendation process. The proposed architecture, as shown in Figure 3.2(a), contains the following components: knowledge base, resource generation, resource ranking, results grouping, and presentation. Additionally, these components are located into three main layers according to the layers of the conceptual architectures described before (Figure 3.2(b)): knowledge base, search and discovery, and user interface and applications.

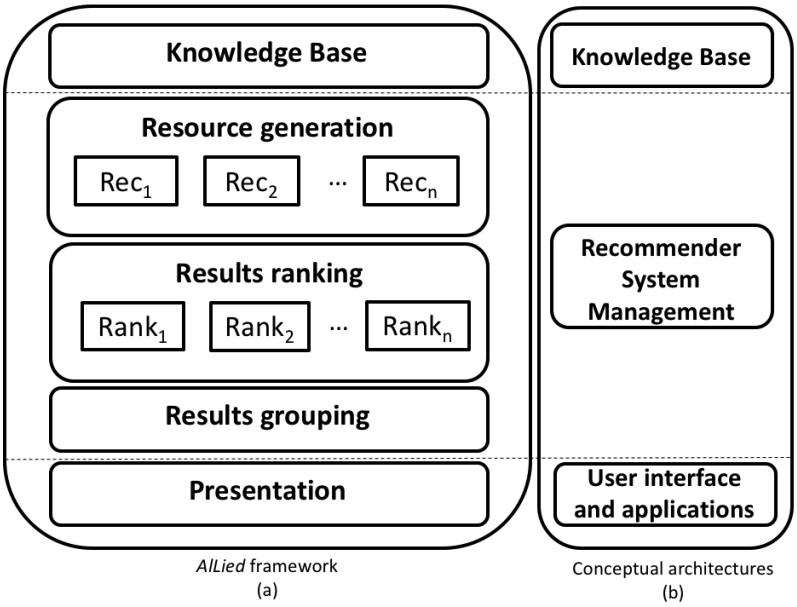


Figure 3.2. Proposed architecture for the *AlLied* framework (a) and its relationships with the common layers of the conceptual architectures for semantic web applications (b).

3.1.1 Knowledge base management layer

This layer is the "data layer" for the *Allied* framework, that provides the interfaces needed to access to local or remote Linked Data datasets. It is traversal to the other layers as it is the main data source containing the knowledge about resources and their structural relationships. Additionally, this layer can access to local copies (dumps) of the Linked Data datasets as well as to remote datasets via their endpoints. Other types of datasets may be derived from more general datasets for example, matrixes with a sub-set of the data extracted from main datasets (e.g., a matrix containing data about films extracted from DBpedia).

The preferred mechanism for describing Linked Data is the RDF language [88] and the most used Linked Data dataset, as demonstrated in Chapter 2, is DBpedia. DBpedia is one of the most general and complete datasets which was even considered as the hub of the Linked Open Data [90].

3.1.2 Recommender System Management layer

This layer provides the mechanisms for retrieving, searching, discovering and ranking resources based on the data extracted from the knowledge bases. This layer contains four main components: dataset manager, resource generation, results ranking, and results grouping.

Resource generation

This layer aims to generate resources related to an initial resource (resources generally are related to a real-world item e.g., a film, a song, a place etc). This layer receives an initial resource (or a set of initial resources) and generates a set of candidate resources located at a predefined distance from the initial resource. Algorithms in this component may or not rank the candidate resources. The most simple algorithm in this component may be a keyword-based search, which extract resources based on the similarity of their names.

Results ranking

This layer mainly ranks (defines which resources are more similar to the initial resource) the candidate resources obtained in the previous layer, based on semantic similarity functions, i.e. the candidate resources generated in the previous layer are sorted according to their semantic distance values with the initial resource.

Results grouping

The *Allied* framework is based on the DBpedia dataset which is a general purpose source of data. For this reason the results obtained contains an inherent ambiguity due the generality of the data used to produce the recommendations. Moreover, a ranked list of recommendations not always is a good way to show the results, because users may require results arranged according to their subjective needs.

Under these circumstances, the results groping component provides mechanisms to categorize the results obtained from the ranking layer arranging the candidate resources into meaningful clusters or contexts elucidating the wide range of categories they belong to.

3.1.3 User interface and applications layer

This layer provides different interfaces to give access to external applications to the recommendation results. In this way the framework *Allied* can easily be integrated into other applications consuming resource recommendations.

It may be noted that each layer may contains multiple algorithms (plugins) that can be used alone or integrated with other algorithms to produce recommendations that may be suitable for different requirements of domains and contexts. For example, Figure 3.2 shows the generation component with a set of algorithms $\{Rc_1, Rc_2, \ldots, Rc_n\}$ that generate resources located at a predefined semantic distance with respect to an initial resource. These algorithms can be integrated with other algorithms of the same layer or the other layers.

Likewise, the algorithms of the ranking component $\{Rk_1, Rk_2, \ldots, Rk_m\}$ may be integrated with the generation component algorithms in order to produce ranked lists based on the semantic relatedness between each tuple (ic, c_i), where ic is the initial resource and c_i is each one of the candidate resources generated by one of the $\{Rc_1, Rc_2, \ldots, Rc_n\}$ algorithms. In this way, it is possible to produce recommendations based on semantic relationships and to study the application of these algorithms under different contexts and conditions.

3.2 Architecture design

In this section the architecture of the AlLied *framework* is decomposed into subsystems which are assigned to the layers proposed in the conceptual architecture presented in Figure 3.2.

3.2.1 Architecture subsystems

The proposed subsystems for the *AlLied* framework are: *KB_Management*, *RS Management*, and *RS Presentation*. Figure 3.3 shows the subsystems of the proposed framework with their relations and required interfaces.

- **KB Management:** it is related with the Knowledge Base Layer as it provides the interfaces needed to access to local or remote Linked Data datasets.
- **RS Management:** it not only provides mechanisms for retrieving, searching, discovering and ranking resources, but also management tasks such as creating new connections to other remote/local datasets. It contains the main components of the RS as it is the central subsystem. It also controls the execution process of the RS.
- **RS Presentation:** this subsystem provides access to the User interface and Applications layer.

3.2.2 Interfaces description

Figure 3.3 also shows the interfaces required for each subsystem:

- **KB Management:**
  - *LocalDataset interface:* it provides mechanisms for accessing to a local dataset. This dataset may be a local RDF store (e.g. a dump of a Linked Data dataset) or a matrix derived from local/remote datasets.

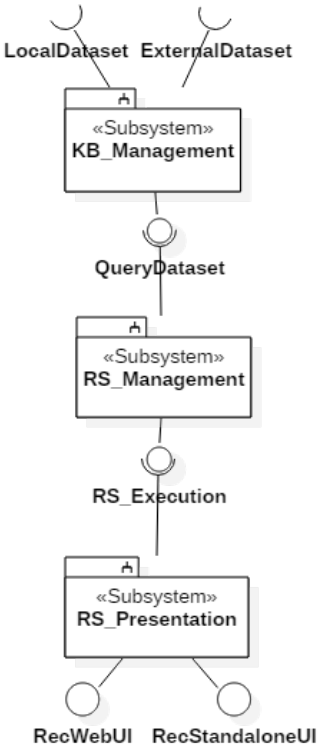


Figure 3.3. Subsystems of the *AlLied* framework

  - *RemoteDataset interface:* it provides access to remote datasets through their endpoints. Endpoints are interfaces to execute SPARQL queries to external datasets.
  - *QueryDataset interface:* this interface allows other subsystems to access query local/remote datasets.

- **RS Management:**
  - *RSExecution interface:* this interface allows other subsystems to control the execution of the recommendation algorithms implemented into the *AlLied* framework.

- **RS Presentation:**
  - *RS_WebUI interface:* it allows web client applications to access to the capabilities of the RS. This may be implemented using a RESTful interface, so the RS may provide access to recommendation algorithms through web services.

– *RS_Standalone interface:* it allows desktop client applications to access to the capabilities of the RS.

3.2.3 Package diagram

The package diagram was developed based on the model MVC (Model - View - Controller). Figure 3.4 shows the main packages of the *AlLied* framework. Three main packages are linked to the root package. 1) The *knowledgebase* package which contains: the model of the RS including the classes for modeling algorithms and datasets that are used by the framework and the *dataaccess* including classes for connecting and querying local/remote datasets. 2) The *control* package which contains classes needed for controlling the execution of the algorithms. 3) The *presentation* package which contains classes to allow client applications for accessing to the RS functionalities.

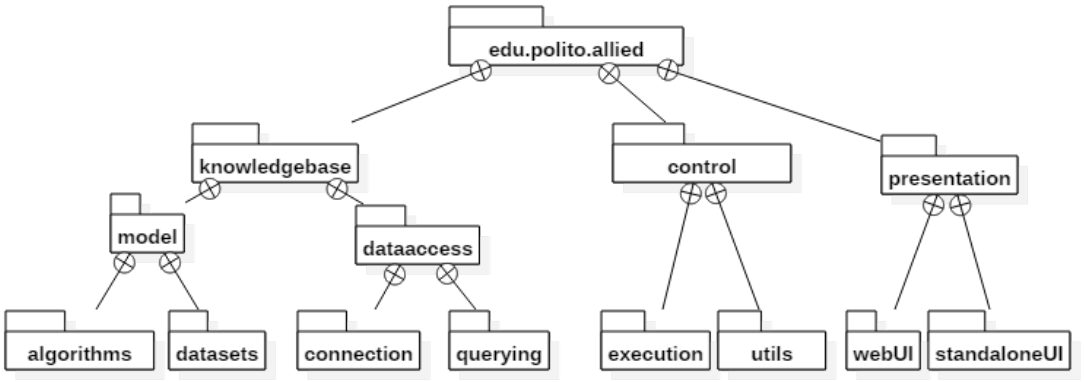


Figure 3.4. Package diagram of the *AlLied* framework

Components of the subsystems

Figure 3.5 shows the main components for each subsystem of the *AlLied* framework.

- **KB Management:**
  - *Query Controller:* this component allows the framework to execute queries on the local/remote datasets.
  - *Category Tree:* the category tree is a hierarchical structure that allows the algorithms to perform operations that are hierarchical-dependent. For example, to compute hierarchical distance, to group resources based on the categories they belong to.

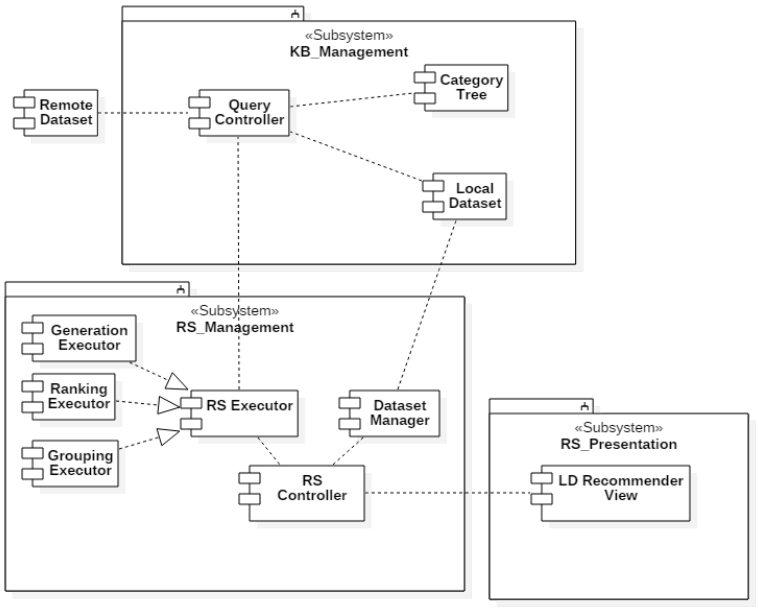


Figure 3.5. Component diagram of the *AlLied* framework

- *Local Dataset:* represents a local dataset for example a dataset dump or a matrix derived from other local/remote datasets.
- *Remote Dataset:* represents a external dataset which is normally accessed through a Linked Data endpoint.

- **RS Management:**
  - *RS Executor:* this component controls the execution of the recommendation algorithms.
    - *Generation Executor:* it implements algorithms for generating candidate resources.
    - *Ranking Executor:* it implements algorithms for ranking candidate resources based on semantic similarity measures.
    - *Grouping Executor:* it implements algorithms for grouping candidate resources based on the category tree.
  - *Dataset Manager:* this component allows the RS Management subsystem to access to the functionalities provided by the KB Management.
  - *RS Controller:* this is a central component of the *AlLied* framework as it controls the access to the recommendation algorithms, as well as to the datasets of the KB Management subsystem.

- **RS Presentation:**
  - *LD Recommender View:* this component embodies both the web user interface and the standalone user interface.

3.2.4 Subsystems Interactions

This section describes the dynamic of the framework through the main sequence diagram to generate candidate resources (CR) or recommendations from an initial resource (aka query resource).

Figure 3.6 shows the general sequence diagram for the recommendation process and the subsystems involved.

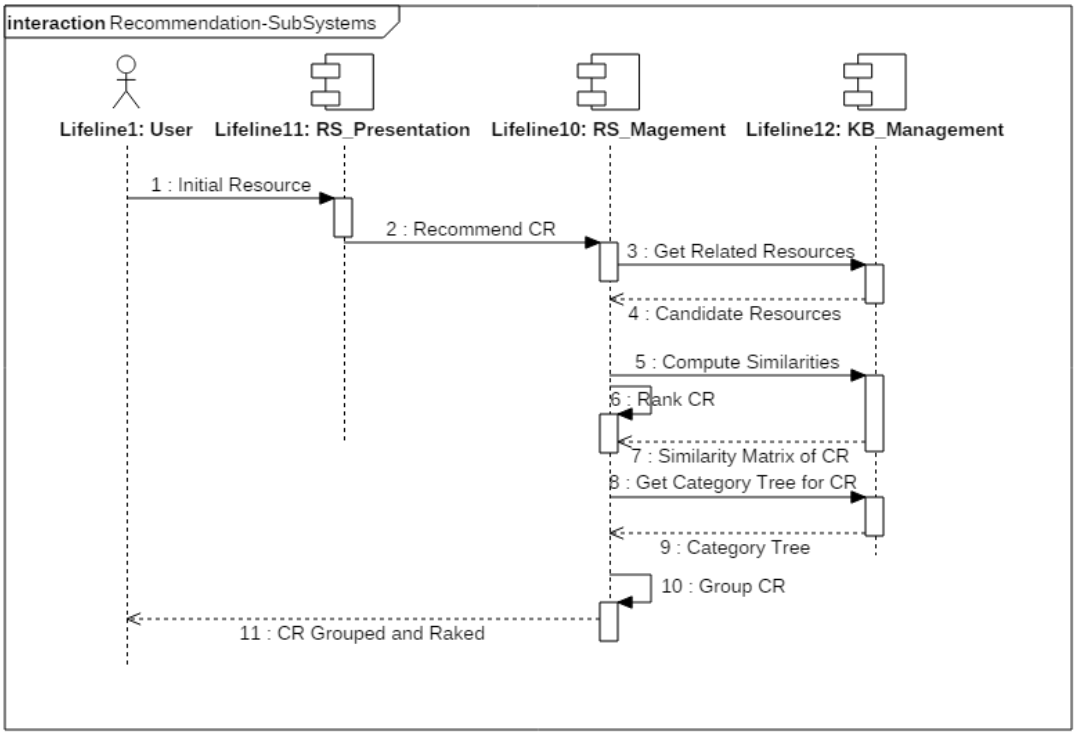


Figure 3.6. Sequence diagram of the recommendation process for the subsystems

Figure 3.6 shows the sequence diagram for the recommendation process and all the components of the subsystems involved.

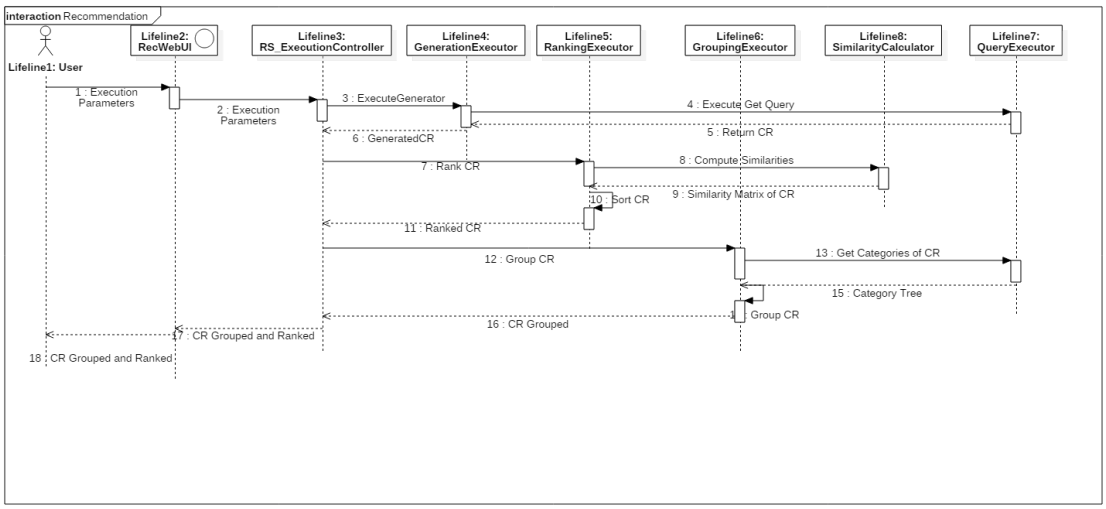


Figure 3.7. Sequence diagram of the recommendation process for the components of the subsystems

3.2.5 Design class diagram

The design class diagram depicts the main classes involved into the recommendation process. Figure 3.8 shows the design class diagram.

- *DatasetConnector:* this class accesses and tests connections to the datasets.
- *QueryExecutor:* this class creates and performs queries to the datasets as required by the RS_Execution class.
- *Algorithm:* this class models a recommendation algorithm. It may be extended as Generator, Ranker, and Grouper classes.
- *RS_ExecutionController:* it executes the algorithms implemented into the *AlLied* framework. Therefor it can execute Generation, grouping, and ranking algorithms.
- *RecommenderView:* this class represents the main component for user interfaces. This class may be decomposed also as WebUI and StandaloneUI as required.

3.2.6 Reference deployment diagram

Figure 3.9 shows the deployment diagram composed of four nodes:

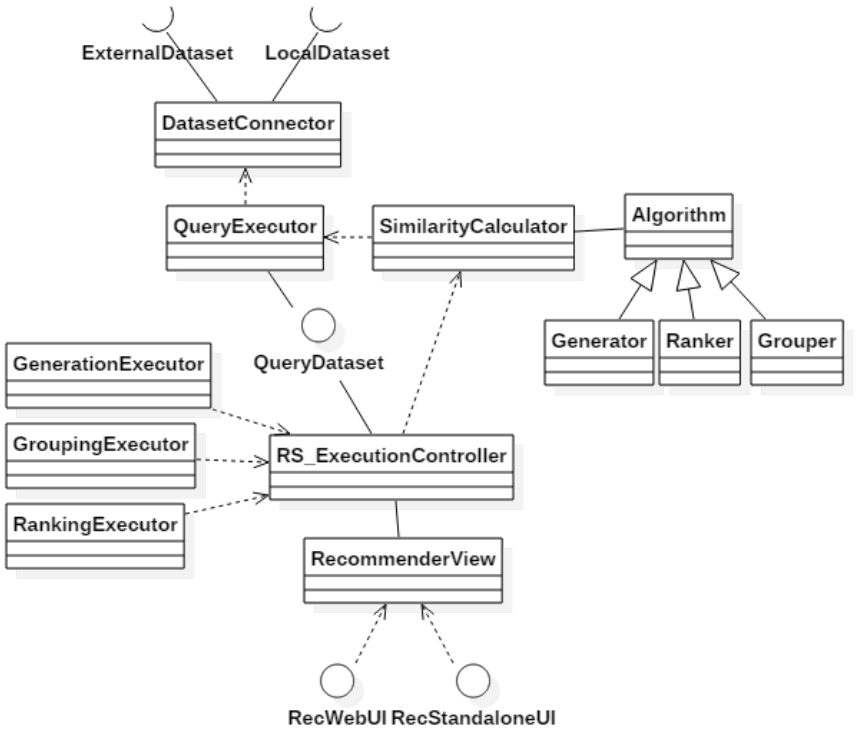


Figure 3.8. Design classes diagram for the *Allied* framework

- *Knowledge base server:* this is a server that stores and provides access and query mechanisms to the local repository, the category tree. Additionally, it may be connected to external datasets, in such a case this server also provides mechanisms for accessing and querying them.

- *Remote Dataset server:* it represents a external server which stores a dataset. Normally, this server may be accessed via a published endpoint.

- *Recommender server:* the recommender server contains the main components of the RS, it controls and executes the recommendation algorithms, and allows them to access and query the knowledge base datasets.

- *Client device:* it is a device for the RS' client. It may be mobile, desktop or web device containing applications suitable to access to the RS.

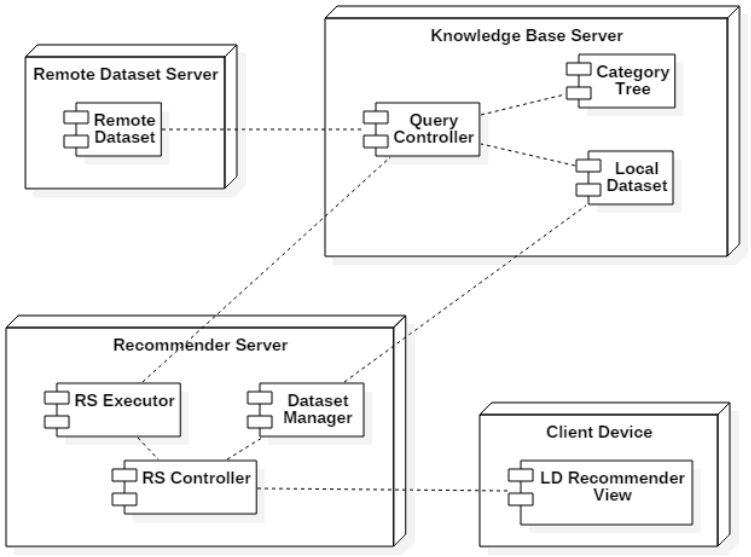


Figure 3.9. Deployment diagram for the *Allied* framework

Chapters 4 and 5 present the implementation of well-know graph-based and machine learning algorithms for each layer respectively. These implementations are useful to study the application of the algorithms in different domains, and to analyze their behavior with different parameters and contexts.

3.3 Summary

This chapter presented *AlLied* a framework to deploy Linked data based algorithms that are useful to generate recommendations. These algorithms may be for generating candidate resources, for ranking them, and for grouping them into meaningful clusters or categories. The main steps for the architectural design of the proposed framework were also presented.

Chapter 4

Allied implementation using graph-based algorithms

This section presents the implementation of graph-based algorithms for each component of the *Allied* framework. The *AlLied* framework is an important contribution of this thesis and its graph-based implementation was described in the research paper entitled "Allied: A Framework for Executing Linked Data-based Recommendation Algorithms"[5], which has been accepted on the "International Journal on Semantic Web and Information Systems". Additionally, this chapter presents other important contribution of this thesis, a new algorithm for Linked Data based resource recommendation named *ReDyAl*. This algorithm was introduced in a conference paper entitled "ReDyAl: A Dynamic Recommendation Algorithm based on Linked Data"[6], which was presented into the 3rd Workshop on New Trends in Content-based Recommender Systems - CBRecSys within the ACM RecSys 2016, which is one of the most important conferences about RS.

Figure 4.1 shows the diagram of the graph-based implementation for the *Allied* framework.

In Figure 4.1 red modules are responsible for the recommendation process, while blue blocks are interfaces for accessing to Linked Data datasets and for presenting the results.

4.1 Knowledge Base Management

4.1.1 Knowledge Base Core

The current implementation for the *Allied* framework uses as knowledge base core a remote dataset named *DBpedia*. However, 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 diverse scenarios. DBpedia is one of the biggest

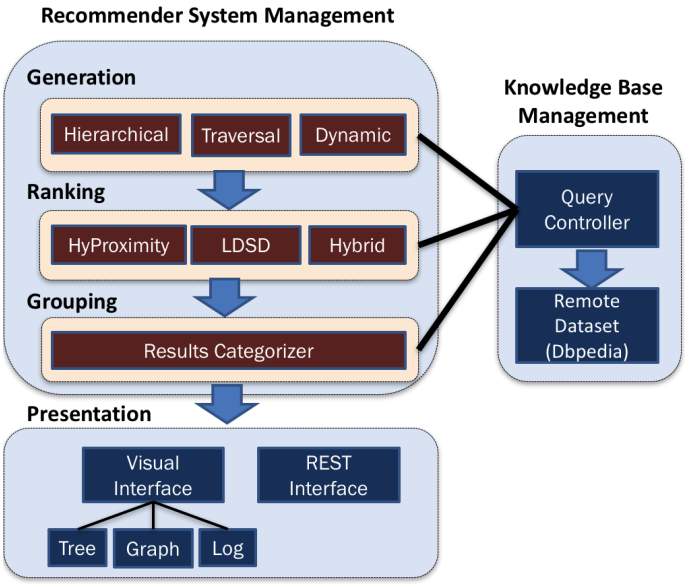


Figure 4.1. Diagram of the graph-based implementation of the *Allied* framework

datasets that is frequently updated as it obtains data from Wikipedia that continuously grows into one of the most interlinked datasets[91].

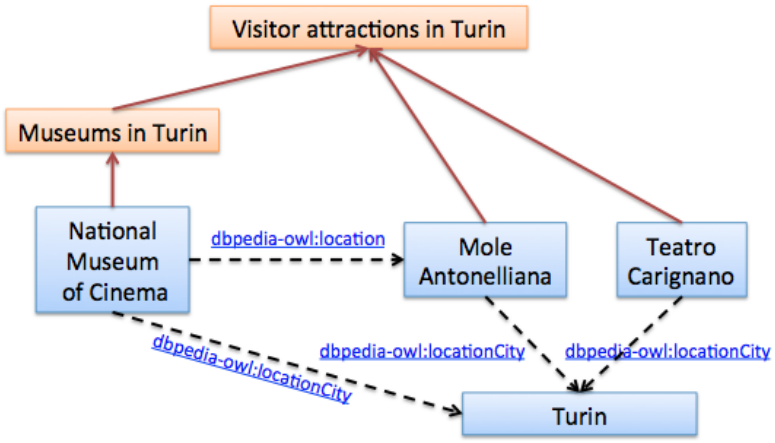


Figure 4.2. Example of hierarchical and traversal relationships in Linked Data

The dataset of the knowledge base may be seen as a tuple $KB- > C, T, R$ composed of concepts (C), categories (T), and relationships (R). For example, in Figure 4.2 concepts are shown in color blue and categories in color orange. The traversal links can be seen with black dotted lines and hierarchical links with plain red lines.

- *Concepts:* are the resources of DBpedia that act as the knowledge item containing the elements of the titles dataset. For example, the resource representing the actress Uma Thurman is:
  <http://dbpedia.org/resource/Turin>
- *Categories:* are the basis of the class hierarchy for the knowledge items. DBpedia provides information about the hierarchical relationships in three different classification schemata:
  - Wikipedia Categories: represented using the Simple Knowledge Organization System (*SKOS*[1]) vocabulary to describe categories and their relations vocabulary. For example, the category of :
    <http://dbpedia.org/resource/Category:Car_manufacturers>
  - YAGO Categories: derived from the Wikipedia category system using WordNet. For example, the category of capitals in Europe:
    <http://dbpedia.org/class/yago/CapitalsInEurope>
  - Word Net Synset Links: generated by manually relating Wikipedia infoboxes and WordNet synsets, and adding a corresponding link to each thing that uses a specific template. For example, the synset of the airlines:
    <http://www.w3.org/2006/03/wn/wn20/instances/synset-airline-noun-2>
- *Relationships:* are links (also known as properties) connecting resources (concepts or categories) along the whole dataset graphs. The knowledge base for the framework contains three types of relationships:
  - Concept-Concept ($C-C$): this type represents the traversal links between concepts. The traversal relationships are those links between resources that are not referred to hierarchical classifications. Most of the links of DBpedia belong to this type.For example, Figure 4.2 shows a direct relationship *<dbpedia-owl:location>* between the concepts "National Museum of Cinema" and "Mole Antonelliana", and an indirect relationship between "Teatro Carignano" and "Mole Antonelliana" through a third concept that in this case is "Turin".
  - Concept-Category ($C - T$): this is the first type of hierarchical relationship that directly links a concept with its parent category. In the SKOS vocabulary this relationship can be identified as: *dcterms:subject* (hasCategory) for representing the relationship concept-category, and *dcterms:subject* (IsCategoryOf) for category-concept. For example, Figure 4.2 shows a concept-category relationship between "National Museum of Cinema" and the category "Museums in Turin".

[1] `http://www.w3.org/2004/02/skos/`

– Category-Category ($T - T$): this is the second type of hierarchical relationships that links categories establishing a hyponymy structure (a category tree). In the SKOS vocabulary this relationship can be found as: *skos:broader* (isSubCategoryOf) to denote broader categories, and *skos:narrower* (isSuperCategoryOf) to denote narrow categories. Figure 4.2 shows this relationship between the categories "Museums in Turin" and "Visitor Attractions in Turin".

In the current implementation for the *Allied* framework, the Wikipedia categories (that use SKOS) were selected because they are the most linked in DBpedia, consisting approximately of 80.9 million links for the year 2014 as reported in [92]. Furthermore, in order to retrieve the data for data from the datasets a submodule for data extraction was developed using the RDF API Jena[2] for java.

4.1.2 Query Controller

This component provides the mechanisms for accessing to the remote dataset (DBpedia). Therefore, this contains a SPARQL query controller, which queries the remote dataset for extracting resources.

4.2 Recommender System Management

This layer contains the main components for executing algorithms for recommendation. Additionally, although no depicted in figure 4.1, it contains the component for controlling the execution of the recommendation algorithms as well as to create combinations (compositions) of them to examine different behaviors of the RS.

4.2.1 Generation component

This layer aims to generate resources related to an initial resource through semantic relationships. This layer receives an initial resource (or a set of initial resources) and generates a set of candidate resources located at a predefined distance from the initial resource. In the graph-based implementation various types of semantic relationships are considered:

Semantic Relationships on Linked Data

There are different ways of classify the relationships existing in DBpedia, for example from the point of view of the RDF framework there are three kind of links depending

[2] `http://jena.apache.org/tutorials/rdf_api.html`

of the type of nodes they involve:

- *resource - literal:* is a type of relationship between a resource and a value such String, numbers and dates. For example a triple from the DBpedia dataset is *<dbpedia:Turin*[3]*, dbpedia-owl:elevation*[4]*, 239.0>* this example shows a triple linking a city as resource and its population as literal.

- *resource - resource:* although formally a literal is also a resource, in this thesis a resource denotes only those containing a URI. For example the triple *<dbpedia:Turin, dbpedia-owl:region, dbpedia:Piedmont>* shows the relationship between Turin as located in the region of Piedmont.

- *resource - blank node:* blank nodes do no identify specific resources, therefore this relationships are intended to say that something with the given relationship exists without explicitly naming it [1].

Other perspective is based on the consideration that resources can be of two types: hierarchical nodes and non-hierarchical nodes. In this thesis, hierarchical nodes are named Categories and non hierarchical nodes as Resources (not even literals or blank nodes). From this point of view the relationships in Linked Data can be classified in hierarchical and traversal. In this thesis, only this type of classification for the relationships is considered.

According to Stankovic et al., [93] these relationships can be:

- *Hierarchical links:* properties that organize resources based on their topics. In Dbpedia these topics can be *types* identified with the prefixes *<rdf:type>* and *<rdfs:subclassOf>*, or *categories <dcterms:subject>* and *<skos:broader>*. It is worth noting that topic, type or category are synonyms the only difference is the categorization schema used and the name that it gives to the topics. In this thesis the name "category" will be used to refer to topics. The hierarchical links can give of two types:

  - *resource - category* when the link connect a subject to its base category. For example the triple *<dbpedia:Turin, dcterms:subject, category:Former_capitals_of_Italy>* shows that Turin belongs to the category of the former capitals of Italy.

[3] *the URI of this resource has been reduced to its prefix for example purposes the full URI is:* `http://dbpedia.org/resource/Turin`

[4] *the full URI of the property is* `http://dbpedia.org/ontology/elevation`

  - *category - category* when the link connects two categories through a relationship of subcategory or broader category. For example the triple *<category:Former_capitals_of_Italy, skos:broader, category:Former_national_capitals>* states that the category of the former capitals of Italy is a subcategory of the broader category former national capitals.

- *Traversal links:* properties that connect resources without establishing a classification or hierarchy. Most of the properties in datasets belongs to this type. For example the triple *<dbpedia:Galileo_Ferraris, dbpedia-owl:deathPlace, dbpedia:Turin>* states that Turin was the death place of the physician Galileo Ferraris.

As can be seen in Figure 4.1, three resource generators were implemented based on the semantic relationships found on the Linked Data.

Traversal generator

The traversal generator is an algorithm that looks for resources directly related with the initial resource and those found through a third resource (indirect relationships). Algorithm 1 is the implementation of the traversal generator used in the *Allied* framework.

Algorithm 1 Traversal generator algorithm

Require: An input URI: $inURI$
Ensure: A set of candidate resources CR
1: $P_{in} = readLinks(inURI)$
2: $FP = getForbiddenLinks()$
3: **for all** $p_k \in P_{in}$ **do**
4: **if** $p_k \in FP$ **then**
5: continue
6: **else**
7: $DCp_k = getDirectResources(p_k)$
8: $ICp_k = getIndirectResources(p_k)$
9: Add DCp_k to CR
10: Add ICp_k to CR
11: **end if**
12: **end for**
13: **return** CR

This algorithm stars reading the links (properties) of an initial resource ($inURI$) and loading a set of forbidden links (Lines - 1 - 2). The set of forbidden links is 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 [94] or nodes that are not desired for the recommendation. In other words, it is a way to limit the results of the algorithm. For example the resource "Turin" contains the link `<dbpprop:populationTotal>` that points to the integer value 911823. Additionally, a set of allowed links may be added in order to restrict the set of resources retrieved to those linked with only a set of specific links.

Afterwards, the algorithm iterates over each link $p_k \in P_{in}$ looking for resources directly connected to the initial resource through the link p_k, and for resources indirectly connected to the initial resource. Optionally, a set of allowed links could be provided in order to restrict the kind of links the algorithm should consider for its execution.

Finally the results are added to the set of candidate resources and returned (Lines 1 - 13). The functions $DCp_k = getDirectResources(p_k)$ and $ICp_k = getIndirectResources(p_k)$ were implemented by executing SPARQL queries using the *jena* API over the standard endpoint of DBpedia[5].

The SPARQL query used to obtain the resources directly connected with the initial resource is presented in Listing 4.1. In this query `<inURI>` is the URI of the initial resources, p is the link and cc is each one of the candidate resources to be retrieved. The forbidden links are limited adding a expression `&& ?p != <forbiddenLinkURI>` for each link.

```
SELECT DISTINCT ?cc WHERE {
        { <inURI> ?p ?cc . }
        UNION{ ?cc ?p <inURI>. }
        FILTER (
                isURI(?cc)
                && ?p != <forbiddenLinkURI1>
                && ?p != <forbiddenLinkURI2>
                && ...
                && ?p != <forbiddenLinkURIn>)). }
```

Listing 4.1. SPARQL query to obtain resources directly linked with the <inURI> resource

The SPARQL query to retrieve resources indirectly connected to the `<inURI>` through a third resource (`o`) is shown in Listing 4.2.

```
SELECT DISTINCT ?cc WHERE {
        { <inURI> ?p ?o .
        ?o ?p ?cc .}
```

[5]The endpoint can be found at: `http://dbpedia.org/sparql`

```
        UNION{<inURI> ?p ?o . ?cc ?p ?o .}
        UNION{ ?o ?p <inURI>. ?o ?p ?cc .}
        UNION{ ?o ?p <inURI>. ?cc ?p ?o .}
        FILTER (
               isURI(?cc) && isURI(?o))
               && ?p != <forbiddenLinkURI1>
               && ?p != <forbiddenLinkURI2>
               && ...
               && ?p != <forbiddenLinkURIn>)). }
```

Listing 4.2. SPARQL query to obtain indirect resources

Again here the forbidden links are limited adding a expression `&& ?p != <forbiddenLinkURI>` for each link.

Hierarchical generator

This module generate candidate resources located at a specified distance in in a hierarchy of categories taken from a category tree described in a Linked Data dataset. For the implementation of this module the category tree of the Wikipedia categories was used.

The algorithm 2 is the implementation of the hierarchical generator module. It starts by creating a category graph (Gc) based on hierarchical information extracted from an initial resource ($inURI$) until reach a maximum level ($maxLevel$) of categories in the category tree of DBpedia. The $maxLevel$ value is used to limit the levels of super categories that the algorithm extract when navigating the category tree (Lines 1 - 7). Those categories are extracted using the hierarchical relationship *skos:broader* from the SKOS model of DBpedia and then the obtained categories are added to Gc (Line 8).

Figure 4.3 shows an example of the category graph for the resource *<http://dbpedia.org/resource/Mole_Antonelliana>*.

Next, the algorithm extracts subcategories $subC_{ij}$ for all the broader categories BC_j found in the last level of the Gc, in order to go one level down to increase the possibility for finding more candidate resources (Lines 9 - 11). Finally the algorithm obtains the resources for each category (including sub-categories) in the Gc, and adds them to the set of categories of the Gc while updating also the edges (Lines 11 - 18).

The functions to obtain broader and sub categories as well to obtain resources for each category were implemented by executing SPARQL queries.

The function $getCategories(URI_{in})$ obtains the set of base categories of the initial resource. Listing 4.3 presents the SPARQL query used where `<inURI>` is the URI of the initial resource.

Algorithm 2 Hierarchical generator algorithm

Require: An input URI: $inURI$, $maxLevel$

Ensure: A category graph Gc containing a ranked set of candidate resources CC

1: $C_{in} = getCategories(URI_{in})$
2: Add C_{in} to Gc
3: **for all** $c_j \in C_{in}$ **do**
4: **while** $level \geq maxLevel$ **do**
5: $BC_j = getBroaderCategoriesUntilLevel(c_j, maxLevel)$
6: **end while**
7:
8: Add BC_j to Gc
9: **for all** $bc_{ij} \in BC_j$ **do**
10: $subC_{ij} = getSubCategories(bc_{ij})$
11: Add $SubC_{ij}$ to Gc
12: **end for**
13: **for all** $c_k \in Gc$ **do**
14: $CC_k = getResources(cg_k)$
15: Add CC_k to Gc
16: **end for**
17: **end for**
18: **return** Gc

```
PREFIX dcterms:<http://purl.org/dc/terms/>
SELECT ?cat WHERE {
       <inURI> dcterms:subject ?cat.}
```

Listing 4.3. SPARQL query to obtain the base categories for the <inURI>

The function $getBroaderCategoriesUntilLevel(c_j, maxLevel)$ recursively extract broader categories for each base category starting from a $level = 1$ until reach the maximum level ($level = maxLevel$). Listing 4.4 presents the SPARQL query used in each iteration. In this query `<catURI>` is the URI of the sub category (c_j). In this query the `FILTER` limits the search for only categories in english language.

```
PREFIX dcterms:<http://purl.org/dc/terms/>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?broaderCat WHERE {
       <catURI> skos:broader ?broaderCat.
       ?broaderCat rdfs:label ?categoryName.
       FILTER (lang(?categoryName) = "en"). }
```

Listing 4.4. SPARQL query to obtain broader categories for the <catURI>

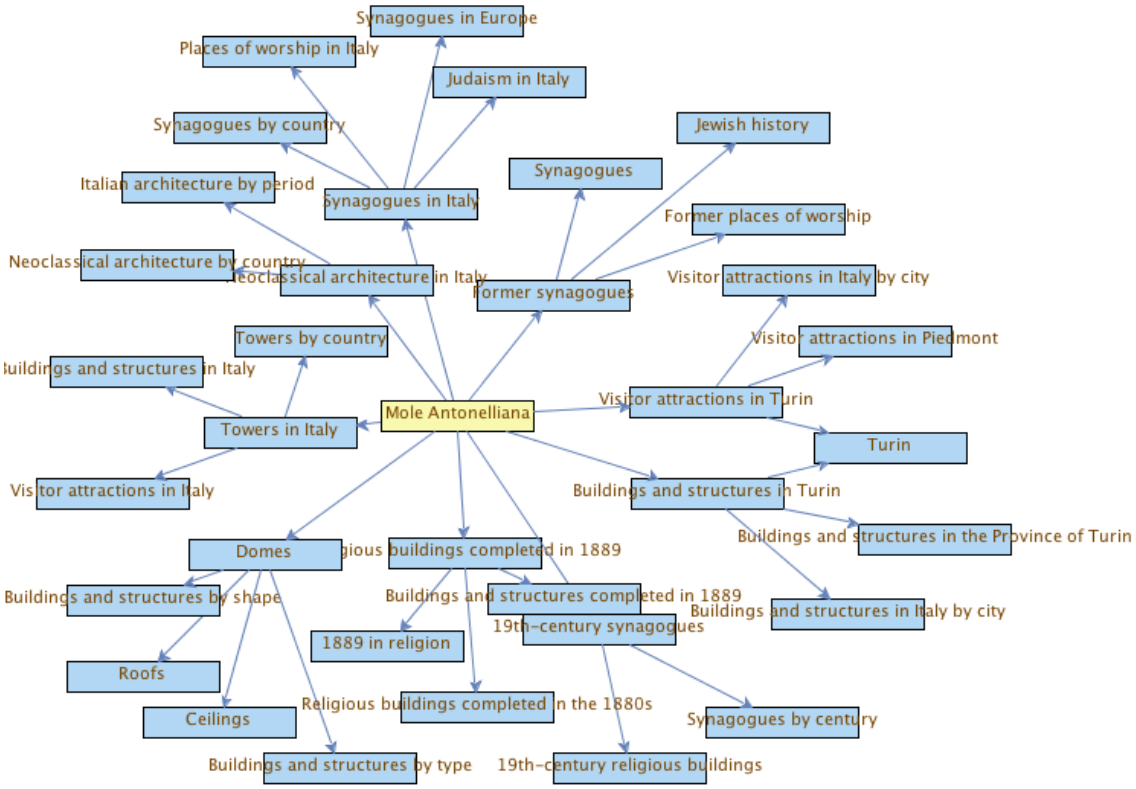


Figure 4.3. Example of the category graph for Mole Antonelliana

The function $getSubCategories(bc_{ij})$ is not recursive, because it only extract sub-categories for each broader category. The set of sub-categories is obtained by the recursive function $getBroaderCategoriesUntilLevel(c_j, maxLevel)$. Listing 4.5 presents the SPARQL query used.

```
PREFIX skos:<http://www.w3.org/2004/02/skos/core#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?subCat WHERE {
                { ?subCat skos:broader <catURI> .
          ?subCat rdfs:label ?label }
UNION { <catURI> rdfs:label ?parent }}
```

Listing 4.5. SPARQL query to obtain the sub categories for the <catURI>

The last function is $getResources(cg_k)$, which obtains the candidate resources for each category ($cg_k \in Cg$) of the category graph generated. Listing 4.6 presents

the SPARQL query used where <catURI> denotes the URI of one of the categories of the category graph to obtain the resource candidates (*cc*).

```
PREFIX dcterms:<http://purl.org/dc/terms/>
SELECT ?cc WHERE {
       ?cc dcterms:subject <catURI> }
```

Listing 4.6. SPARQL query to obtain the subjects of the category <catURI>

Dynamic Algorithm (*ReDyAl*)

In this section a new algorithm for Linked Data based resource recommendation is proposed. This algorithm is a contribution of this thesis, it was introduced in a conference paper entitled "ReDyAl: A Dynamic Recommendation Algorithm based on Linked Data"[6], which was presented into the 3rd Workshop on New Trends in Content-based Recommender Systems - CBRecSys within the ACM RecSys 2016, which is one of the most important conferences about RS.

Interlinking is one of the four principles to publish data as Linked Data on the Web, it further makes possible to discover more related resources. The algorithm presented in this section, named *ReDyAl*, proposes a dynamic strategy that can be divided in three stages:

- The first stage find resources analyzing the interlinking or number of connections that an initial resource contains versus other resources.
- The second stage analyzes the classification of resources based on categories and found similar resources located in common categories.
- The third stage intersect the the results of both stages given priority to the interlinking.

Thanks to this dynamic nature which considers the different relationships between resources, *ReDyAl* is useful for those cases dealing with datasets where there may be "well-linked" resources as well as poor linked resources. In these cases the dynamic algorithm is able to choose the best strategy to find candidate resources to be recommended based on the implicit knowledge contained in the Linked Data relationships.

Additionally, the *ReDyAl* algorithm may be configured with a set of forbidden links and allowed links in order to restrict the kind of links the algorithm should consider.

Algorithm 3 Dynamic Generator Algorithm *(ReDyAl)*

Require: $inURI$, $minT$, $minC$, FP, $maxLevel$
Ensure: A set of candidate resources CR

```
1: P_in = readAllowedLinks(inURI, FP)
2: if |P_in| ≥ minT then
3:    for all p_k ∈ P_in do
4:       DCp_k = getDirectResources(p_k)
5:       ICp_k = getIndirectResources(p_k)
6:       Add DCp_k to CR_tr
7:       Add ICp_k to CR_tr
8:    end for
9:    if |CR_tr| ≥ minC then
10:      return CR_tr
11:   else
12:      currentLevel = 1
13:      Gc_in = createCategoryGraph(URI_in, currentLevel);
14:      while currentLevel ≤ maxLevel 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:         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
```

The *ReDyAl* algorithm (Algorithm 3) receives as input an initial resource represented as an initial URI (URI), and three values ($minT$, $minC$, $maxLevel$) for configuring its execution: $minT$ is the minimum number of links to consider that a resource is "well linked", i.e, an algorithm user can define this value to tell the algorithm to prioritize the interlinking to generate the candidate resources, $minC$ is the minimum number of candidate resources that the algorithm is expected to generate, and $maxLevel$ is only to limit the number of levels in the category tree that the algorithm could consider. This later value may be defined manually and it is useful when a resource does not contain links to the category tree and it could not be possible to generate a category graph. Additionally, the algorithm may receive a list of "forbidden links" to limit the search of candidate resources over a predefined

list of undesired links that can be specified manually.

(ReDyAl) algorithm starts by obtaining 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. If there is a considerable number of allowed links, i.e., the initial resource is well-interlinked then the algorithm obtains a set of candidate resources located through direct or indirect links 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 and if these are greater than or equal to the $minC$ then the results are considered enough and are returned. (Lines 9 - 10) Otherwise the algorithm generate a category graph with categories of the first level and applies iterative updates over the category graph over n levels above the initial resource until at least one of two conditions is fulfilled: the number of candidate resources is enough ($CR\ leqMinc$), or the maximum number of levels is reached ($CurrentLevel\ geqmaxLevel$) (Lines 14 - 23). In any case the algorithm combines this results with the results obtained in the Lines 3 to 8.

4.2.2 Ranking component

The *Allied* framework in its current graph-based implementation includes (but is not limited) four ranking algorithms. Similarly to the algorithms of the generation layer, the ranking algorithms are also based on the semantic relationships and the corresponding similarity measures for Linked Data.

Graph-based similarity Measures for Linked Data

AlemZadeh [95] shows a classification of the similarity measures taking into account the hierarchical and traversal relationships: In the following, these measures are described.

- *Hierarchical measures (category-to-category):* is the semantic distance between two categories represented as the number of leaps over the category graph to go from one category to the other.
- *Traversal measures (resource-to-resource):* is the distance calculated as the shortest path between two resources. However in this case the resources are connected trough different types of links. In the case of DBpedia, for example, these links can be redirects, disambiguation links, wikilinks, and properties

Next the most important measures for both classifications are studied.

Passant [18] defined a Linked Data Semantic Distance *(LDSD)* between two resources published on a Linked Data dataset. In this measure the similarity of two resources (c_1, c_2) is measured combining four properties: the input/output direct

links or the input/output indirect links between them. The Equation 4.1 is the basic form of the *LDSD* distance.

$$LDSD(c_1, c_2) = \frac{1}{1 + Cd_{out} + Cd_{in} + Ci_{out} + Ci_{in}} \tag{4.1}$$

Where Cd_{out} is the number of direct input links (from c_1 to c_2), Cd_{in} is the number of direct output links, Ci_{in} the number of indirect input links, and Ci_{out} the number of indirect output links.

Stankovic et al. [93] defined a semantic similarity measure known as *HyProximity* that is based on the structural relationships that may be inferred from resources of a Linked Data dataset. The *HyProximity* in its general form is shown in Equation 4.2 as the inverted distance between two resources, balanced with a pondering function.

$$hyP(c_1, c_2) = \frac{p(c_1, c_2)}{d(c_1, c_2)} \tag{4.2}$$

In this equation $d(c_1, c_2)$ is the distance function between resources c_1 and c_2 and $p(c_1, c_2)$ is the pondering function that is a weight function used to give a level of importance to different distances. Based on the structural relationships (hierarchical and traversal) different distance and pondering functions may be used to calculate the *HyProximity* similarity.

For the hierarchical relationships the distance $d_h(c_1, c_2)$ may be seen as the shortest path-based from c_1 to the first common ancestor (category) that it shares with c_2 and the pondering function can be the informational content of the closest common category. The informational content is a measure that combines statistical information with the hierarchical structure of a category tree, usually determined by a higher level category that subsumes two resources [96]. The informational content may be calculated as the probability $p(C)$ of encountering a category C in the category tree, so in categories of high levels the informational content is lower.

For the traversal relationships the distance $d_{trav}(c_1, c_2)$ is assumed to 1 for each link (direct or indirect) that exists between two resources over one of a set of predefined transversal properties. In this case Stankovic et al., selected manually a set of relevant traversal properties for a specific domain. The pondering function $p_{trav}(c_1, c_2)$ is calculated as function of the relationship between the number of resources (n) connected over a specific property and the total number of resources of the dataset (M) (Equation 4.3):

$$p_{trav}(c_1, c_2) = -\log \frac{n}{M} \tag{4.3}$$

Additionally Stankovic et al., defined a mixed distance function which assigns the value n for the resources that both the hierarchical and traversal functions found

at level n.

$$HyP_{hybrid}(c_1, c_2) = HyPtrav(c_1, c_2) + HyPhi(c_1, c_2) \tag{4.4}$$

Traversal LDSD Ranking

This ranking algorithm is traversal as it calculates the Linked Data Semantic Distance *(LDSD)* between the initial resource and each one of the candidate resources obtained in the generation layer. The LDSD distance, initially proposed by Passant [18], is based on the number of indirect and direct links between two resources (Equation 4.1). Unlike the implementation developed by Passant that is limited to links from a specific domain, the LDSD function implemented in *Allied* takes into account all the resources of the dataset. However, it can be limited adding a set of Forbidden links to be customized to defined types of links belonging or not to a specific domain.

The SPARQL query that counts the input and output direct links between the initial resource (*<inURI>*) and a resource of the set of candidate resources (*<ccURI>*) is presented in Listing4.7:

```
SELECT count(?p) WHERE {
       #output links
       { <inURI> ?p <ccURI> . }
       #input links
       UNION
       { <ccURI> ?p <inURI>. }
}
```

Listing 4.7. SPARQL query to count input and output direct links

The SPARQL query that counts the input and output indirect links between the initial resource (*<inURI>*) and a resource of the set of candidate resources (*<ccURI>*) is presented in Listing 4.8:

```
SELECT count(?p) WHERE {
       #input links
       {?o ?p <inURI> . ?o ?p <ccURI> .}
       UNION
       {?o ?p <inURI> . <ccURI> ?p ?o .}
       #output links
       UNION
       {<inURI> ?p ?o . ?o ?p <ccURI> .}
       UNION
       {<inURI> ?p ?o . <ccURI> ?p ?o .}
```

```
}
```

Listing 4.8. SPARQL query to count input and output indirect links

Using these two SPARQL queries the traversal ranking algorithm calculates the LDSD for each pair of resources composed of the initial resource and each one of the resources obtained from the generation layer.

HyProximity Ranking

This algorithm is based on the *HyProximity* measure (Equation 4.2) defined by Stankovic et al. [93], which can be used to calculate both traversal and hierarchical similarities:

- *HyProximity hierarchical (*$hyProximity_{hierarchical}$*)*: as stated in Section 4.2.2, this similarity is the quotient of a pondering function (p) and a distance (d). The distance was calculated using the maximum level of categories of the hierarchical generator algorithm (Algorithm 2) such that: $d(ic, c_i) = maxLevel$, where ic is the initial resource and c_i is each one of the candidate resources generated in the hierarchical algorithm. The pondering function was calculated with an adaptation of the informational content function (Equation 4.5) defined by Seco et al. [97]. In this equation $hypo(C)$ is the number of descendants of the category C and $|C|$ is the total number of categories in the categoryGraph.

$$p(C) = 1 - \frac{\log(hypo(C) + 1)}{log(|C|)} \tag{4.5}$$

  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 [98].

- *HyProximity traversal (*$hyProximity_{traversal}$*)*: in this similarity function the distance $d(ic, c_i) = maxLevel$ if the generator of resources is hierarchical, otherwise $d(ic, c_i) = 1$ for resources connected to the initial resource through direct traversal links or $d(ic, c_i) = 2$ for indirect traversal links. The ponderation function in this case was calculated with the equation 4.3, which is the quotient of the number of the candidate resources produced in the generation layer ((n)) connected over a specific link and the total number of resources of the dataset ((M)).

  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 Section 4.2.1. Accordingly, the SPARQL queries shown in Listings 4.7 and 4.8 may be used to compute the number of direct and indirect links. The value of M was fixed to the value of 4 584 616 which is the number of "things" contained in the *DBpedia* dataset according [92].

Hybrid Ranking

This ranking integrate the traversal *LDSD* or traversal *HyProximity* techniques with the hierarchical algorithm. It is worth noting that due the *LDSD* is a distance measure, in order to be combined with one of the *HyProximity* similarities may be transformed to a similarity measure too. Additionally, in this ranking, two weight parameters are defined in order to give more or less importance to the hierarchical or traversal ranking. In this way the hybrid ranking uses the Equation 4.6, where α is the weight for the traversal algorithm and β is the weight for the hierarchical algorithm.

$$Hybrid_{sim} = (1 - LDSD)\alpha + (hyProximity_{hierarchical})\beta \tag{4.6}$$

In this equation the expression $1 - LDSD$ may be changed by the function $hyProximity_{traversal}$.

4.2.3 Grouping component

In the current implementation of the *Allied* framework there is one approach to categorize the results based on the hierarchical relationships of the Linked Data cloud. In this way, when an application requires to classify resources according to an application domain the grouping algorithm provides a mechanism to access easier to recommended items organized by broader categories which an also be considered as explanations for the recommendations. Algorithm 4 is the implementation for the grouping layer.

The Algorithm 4 receives as input a set of ranked candidate resources(CR), an initial resource $inURI$, and optionally an initial category graph (Gc_{in}). If Gc_{in} is not given then the algorithm creates a new Gc for the initial resource and until a level $maxLevel$, otherwise the algorithm creates a copy of Gc_{in}. In this implementation a $maxLevel = 2$ was selected because at this value it was possible to obtain a reasonable relationship between the number of categories and the time consumed.

Afterwards, the algorithm extracts the categories of the highest level ($C_{maxLevel}$) and creates pairs of categories combining the elements of $C_{maxLevel}$. Next the function $getLessCommonBroaderCategory(c_i, c_j)$, which is based on the less common ancestor, is executed to find a set of broader categories subsuming the categories of the set $C_{maxLevel}$.

These categories are intersected and a function *deleteEmptyCategories* is executed which remove from the graph those categories subsuming less than three

Algorithm 4 Hierarchical Grouping Algorithm

Require: CC, $inURI$, optionally Gc_{in}
Ensure: A category graph Gc
1: **if** $Gc_{in} = null$ **then**
2: $Gc = createCategoryGraph(maxLevel)$
3: Add CR to Gc
4: **else**
5: $Gc = Gc_{in}$
6: **end if**
7: $C_{maxLevel} = getMaxlevelCategories(Gc)$
8: **for each** pair of categories $(c_i, c_j) \in C_{maxLevel}$ **do**
9: $c_{lcb} = getLessCommonBroaderCategory(c_i, c_j)$
10: Add c_{lcb} to Gc
11: Add $edge(c_i, c_lcb)$ and $edge(c_j, c_lcb)$ to Gc
12: **end for**
13: $intersectCategories(Gc)$
14: $deleteEmptyCategories(Gc)$
15: **return** Gc

subcategories (i.e. only the categories c_i, c_j). In this way a grouping of higher level for the candidate resources is created.

4.3 Presentation

The current implementation of *Allied* include two main interfaces that provides mechanisms to present the results to the final user.

4.3.1 RESTFul Interface

This interface is represented through a (Representational State Transfer) RESTFul Web service that provides web-based operations to access to the different algorithms implemented along the layers of the *Allied* framework. The RESTful services are based on a lightweight architectural style on top of the HTTP protocol, allowing the framework to expose the functionalities by methods in the HTTP standard (GET, PUT, POST, DELETE) based on a uniform interface. Additionally, this interface allows the framework to present the results in various formats such as *HTML*, *XML* or *JSON*.

In this way, applications may access to the functionalities of the recommender via web service methods that offers functionalities as for example: generate resource

recommendations from an initial resource, create category graphs and obtain candidate resources for each category, execute ranking algorithms to sort results of a generator algorithm, among others. For example a mobile application that is currently accessing to this framework via the RESFul interfaces is being developed in collaboration with Telecom Italia in order to generate films recommendations. Appendix C.1 contains the graphical interface of this application where a "mind map" is presented based on the information provided from the recommender. Additionally, Appendix C.3 shows an example of a Web Application developed over the *AlLied* framework.

4.3.2 Standalone Interface

The standalone interface is intended to allows desktop applications: to execute the algorithms of the framework, to obtain intermediate results for each phase of the recommendation process (generation, ranking, and grouping), and to view the results in different formats. The current implementation include an interface to show the results integrated into a JTree, and a graph view.

The JTree view shows the information into a hierarchical organization as a tree structure similar to folders in a file system. This interface allows the users to browse the categories from the more general categories to the most specific. Unlike the JTree view that allows to navigate through the categories but hides the link structure of the recommendations, the graph view is intended to show the different relationships between the candidate resources, the initial resource and the surrounding categories.

Appendix C.2 presents the user interfaces developed for the standalone interface. Additionally, Appendix C, presents an example of the candidate resources for the initial resource $<http://dbpedia.org/resource/Mole\_Antonelliana>$ in a folder system structure (Section C.2.1), as well as the graph structure view (Appendix C.2.2).

4.4 Summary

This chapter presented a graph-based implementation of the *AlLied* framework. Additionally, a dynamic algorithm for resource generation (recommendation) named *ReDyAl* was proposed. This algorithm dynamically analyzes the knowledge obtained from relationships between resources as well as the categorization environment where these are classified, giving priority to the interlinking.

Chapter 5

Allied implementation using machine learning algorithms

This chapter describes the creation of a Linked Data based dataset and its implementation and configuration of machine learning algorithms within *Allied*.

The main idea of using machine learning algorithms in a RS based on Linked Data is to obtain a different perspective of algorithms which are not based on the intrinsic graph structure of the Linked Data datasets but on regularities and patterns in data. These patterns may be useful for predicting unknown similarities between items in order to produce recommendations.

Nowadays, there is a large variety of machine learning algorithms developed in software packages like Weka, R and RapidMiner. These toolkits include a vast set of algorithms for various task in data mining applications. This section presents the selection and use of these algorithms for each layer of the Allied framework presented in Fig. 3.2.

Figure 5.1 shows the machine learning algorithms included for the current implementation of the Allied framework. These algorithms where selected taking into account the most used algorithms into the state of the art as described in section 2.3.5 of the State of the Art. However due to the *AlLied* framework is extensible it is also possible to implement other algorithms.

5.1 Knowledge Base Management

5.1.1 Knowledge Base Core

In data mining it is well-known that the quality of the results obtained by a machine learning algorithm con only be as good as the data they get as input [99]. In other words, the construction of a dataset, suitable for machine learning algorithms, is a crucial step.

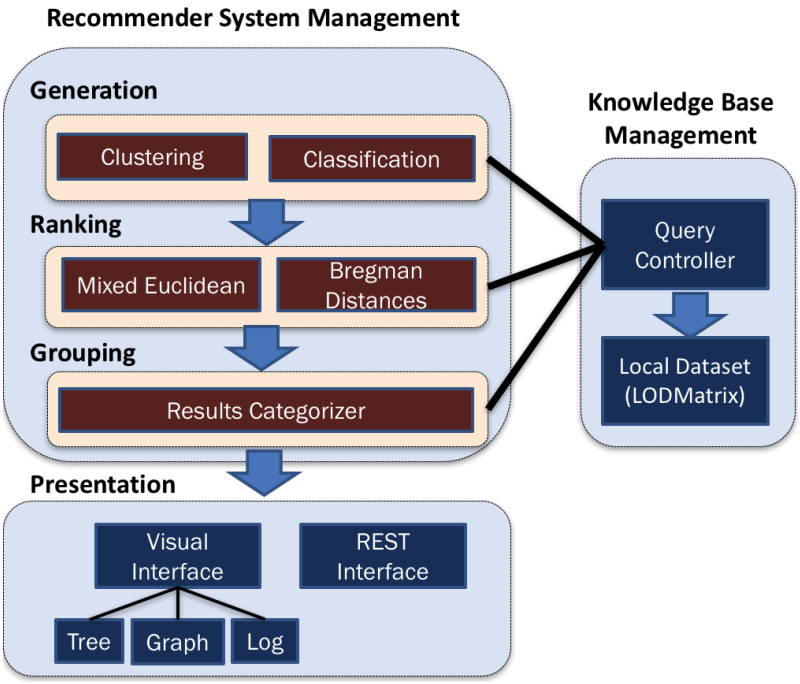


Figure 5.1. Machine learning algorithms implemented into the *Allied* framework

Dataset development

Even though the knowledge base core for the graph-based implementation is a Linked Data dataset such as the DBpedia dataset, in this implementation other type of datasets suitable for machine learning algorithms were developed. The construction of this dataset followed a set of steps to ensure quality in knowledge discovery tasks as described into the conceptual framework *FDQ-KDT* [100].

The *FDQ-KDT* framework was developed to address poor quality data in knowledge discovery tasks. DBpedia is a dataset collaboratively built as it is based on Wikipedia, so its quality is poor because it is prone to errors that human collaborators can commit when they enter information on wikipedia. Hence, the *FDQ-KDT* framework is just suitable to ensure quality to build a new dataset derived from DBpedia.

The execution process of the *FDQ-KDT* framework establishes the following phases: data fusion for combining data from different sources; data quality diagnosis in order to describe, explore and asses the data quality; data selection for fixing problems found in the previous phase; and data construction, where new information is generated based on specific machine learning tasks.

Accordingly, in the construction of the dataset for the knowledge base core layer the following steps were conducted.

1. **Data extraction:** first at all data from the Linked Data dataset are extracted.

As stated before, for simplicity the only dataset used in the present proposal is DBpedia, so the data fusion phase is not needed.

The main drawback of using machine learning algorithms rather than graph-based algorithms for RS is that the data extraction step need to know a priory the application domain in order to obtain only relevant data of that domain. For the current implementation, the movies domain was chosen. Therefore, a SPARQL query to obtain the sub-set of instances that represent films need to be executed. Listing 5.1 shows the SPARQL query used.

```
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?film WHERE
{ ?film a dbpedia-owl:Film. }
```

Listing 5.1. SPARQL query to obtain films from DBPedia

The result of the query presented on Listing 5.1 was a list of 106613 URIs of films obtained from DBpedia. Each URI obtained represents a film, which contains data as well as properties and semantic links to other URIs (which may be or not films). As most of the machine learning algorithms accept as input a dataset represented as a matrix, a database and various matrices derived from it were created. In these matrices rows represent films and columns attributes of these films. Attributes[1] are the properties or links for each film.

Until here the result obtained from Listing 5.1 is a list of URIs. However, properties and links are needed to be extracted iteratively from the whole list of URIs in order to build matrices suitable for machine learning algorithms. Taking into account that films extracted from DBpedia contain a huge number of properties, much of them belonging to only one film and others not relevant for the recommendation problem, it is necessary to choose a sub-set of the most significant properties.

- *Attribute Selection*:

  Accordingly, a review of the main RS (from research and from industry) related with films recommendations was conducted. The details of this review, as well as the selected RS and their references are found in Appendix D.1. As a result of this review, for each RS a list of the attributes that were used for recommendations was extracted. The attributes extracted and their frequency of occurrence in RS are shown in Table 5.1.

[1] Attributes and properties are hereinafter referred indistinctly to features of films represented as links in the Linked Data dataset

Feature	Frequency
subject - category - genre	7
starring - actor	7
country	6
director	6
writer	5
producer	5
year - date	4
music composer	3
distributor	3
runtime	2
type	1
title	1
soundtrack	1
ratings	1
performance	1
narrator	1
language	1
editor	1
cinematography	1

Table 5.1. Attributes extracted and their frequency of occurrence in RS

As shown in Table 5.1, attributes with different names but referring to the same resource were grouped, e.g. subject, category and genre. Furthermore a sub-set of the most common attributes among all the RS was selected as the set of attributes considered for building the matrices were later used as inputs to the machine learning algorithms. The attributes selected were those with a frequency greater than 3, i.e. *subject, starring, country, director, writer, producer, and year.*

Once the set of attributes was selected, the next step was extracting their corresponding values for each film from DBpedia. To this end, the SPARQL query presented in Listing 5.2 was executed for each film restricted to the attributes selected.

```
SELECT DISTINCT ?property ?object WHERE {
{
        <filmURI> ?attribute ?value.}
        UNION
        {?value ?attribute  <filmURI>}
        FILTER
        (?attribute = <allowedAtt1>
        || ?attribute = <allowedAtt2
        || ... || ?attribute = <allowedAttN>). }
```

Listing 5.2. SPARQL query to obtain attribute values for each film

The result of the SPARQL query shown in Listing 5.2 is a list of attributes (*?attribute*) and their values (*?value*) for a film represented by its URI (<filmURI>). The attributes <allowedAtt1> ... <allowdAttN> are a list of attributes allowed for the SPARQL query, i.e. the list of selected attributes that were taken into account for building the matrices. For

example, some of the pairs attribute-value for the film named *Infernet* (<http://dbpedia.org/resource/Infernet>) are shown in Table 5.2.

Attribute	Value
country	Italy
name	Infernet@en
director	Giuseppe_Ferlito
starring	Ricky_Tognazzi
starring	Elisabetta_Pellini
starring	Remo_Girone
starring	Katia_Ricciarelli
starring	Daniela_Poggi
starring	Roberto_Farnesi
starring	Andrea_Montovoli
starring	Giorgia_Marin
starring	Laura_Adriani
writer	Roberto_Farnesi
subject	CategoryItalian-language_films
subject	CategoryItalian_films
subject	2015_films

Table 5.2. Example of some of the pairs attribute-value for the film *Infernet*

- *Matrix generation*:

  As shown in Table 5.2 some attributes contain more than one value, therefore there are different approaches for building matrices. One approach is to build a matrix where each row is a film and the values for each attribute is the number of times a film contains that attribute; other approach is to build a binary matrix where instead of the number of times of each attribute a value 1 is set when a film contains it and 0 otherwise; and a more complete approach is to create various rows for each film in order to represent all the possible combination of value-attribute for each film.

  Although in this thesis, various matrices for these approaches were created, the most complete one (containing all the possible combination of value-attribute) was selected and named as *LODMatrix*. The rest of matrices, that were developed for future works, are described in Appendix D.2. Additionally, a database named lodmatrixdb was developed to ease the creation of various types of matrices. The description of the lodmatrixdb is also shown in Appendix D.2.

  The *LODMatrix* contains approximately 10100 films (about 3 million rows, considering that each film is represented by multiple rows) with 8

properties (the 7 properties selected before, and the name of the film). For example, Table 5.3 shows an example of the film entitled "Infernet", which contains only 1 country, 1 director, 2 producers, 2 writers, and 3 subjects, so the number of rows is $1 \times 1 \times 2 \times 3 \times 3 = 12$. In this way, all the data for the selected attributes is represented. Note that other films may contain until hundreds of rows.

Name	Country	Directors	Producers	Starring	Writers	Year	Subjects
Infernet	Italy	Giuseppe Ferlito	Federica Andreoli	11	Marcello Iappelli	2016	2015 films
Infernet	Italy	Giuseppe Ferlito	Federica Andreoli	11	Marcello Iappelli	2016	Italian films
Infernet	Italy	Giuseppe Ferlito	Federica Andreoli	11	Marcello Iappelli	2016	Italian language films
Infernet	Italy	Giuseppe Ferlito	Federica Andreoli	11	Roberto Farnesi	2016	2015 films
Infernet	Italy	Giuseppe Ferlito	Federica Andreoli	11	Roberto Farnesi	2016	Italian films
Infernet	Italy	Giuseppe Ferlito	Federica Andreoli	11	Roberto Farnesi	2016	Italian language films
Infernet	Italy	Giuseppe Ferlito	Michele Calì	11	Marcello Iappelli	2016	2015 films
Infernet	Italy	Giuseppe Ferlito	Michele Calì	11	Marcello Iappelli	2016	Italian films
Infernet	Italy	Giuseppe Ferlito	Michele Calì	11	Marcello Iappelli	2016	Italian language films
Infernet	Italy	Giuseppe Ferlito	Michele Calì	11	Roberto Farnesi	2016	2015 films
Infernet	Italy	Giuseppe Ferlito	Michele Calì	11	Roberto Farnesi	2016	Italian films
Infernet	Italy	Giuseppe Ferlito	Michele Calì	11	Roberto Farnesi	2016	Italian language films

Table 5.3. Example of the film *Infernet* from the LODMatrix

2. **Data quality assessment:** Due to DBpedia contains a lot of missing or wrong data, the second step is the data quality diagnosis. This step identifies issues on the data such as outliers, incompleteness, and timeliness. Outliers are observations which deviate so much from other observations or the lack of harmony between different parts or elements; incompleteness is referred to missing values; and timeliness is the degree to which data represent reality from the required point in time.

In this implementation of the Allied framework, only the outliers were considered. Even though, the LODMatrix contains a lot of missing values, techniques for solving the problem of incompleteness such as imputation data are only suitable for numerical data which may be predicted or computed using statistical approaches; methods like hot deck imputation can't be applied because the range of values for columns with missing values is not fixed. Timeliness was not considered because it is data that despite it does not contain any of the other issues described above, represent a different behavior because is out of the current time line (for example, very old data). However in a RS for films timeliness is not a problem because even old films may be of the interest of a user.

The *FDQ-KDT* framework [100] also lists the approaches that may be used in order to address each one of the data quality issues. Accordingly, two approaches for outlier detection were used: local outlier factor (LOF) and Tukey's method which uses interquartile (IQR) range approach. The description of the execution of these approaches on the LODMatrix is described in Appendix D.3.

The results of this step for the quality diagnosis were useful for detecting films that contained erroneous data or extreme values. However, as the LODMatrix was developed in order to use machine learning algorithms for recommending films, no films should be removed as every film is a potential candidate for recommendation depending on the user's query. For this reason, the films containing outliers and extreme values were not removed from the dataset but corrected in order to contain consistent values. Part of the process of data correction of the LODmatrix was carried out manually, however considering that this dataset contains more than 3 million rows it represented a demanding and tedious task. For this reason, other approaches were considered in order to retrieve data to fix erroneous data as well as missing values.

Accordingly, two free web services were employed in order to accomplish this task. The first one is *The Movie DB (TMDb)*, which may be used to make HTTP requests to obtain JSON formatted information about films or TV shows; and the second one is *The Open Movie Database (OMDb)*, which is similar to the first one, but it is based on the popular *Internet Movie Database (IMDb)* that contains information related to films, television programs and video games. The OMDb also accepts HTTP requests and returns JSON or XML formatted responses. Both web services are continuously updated because they are contributed and maintained by their users.

An application in the Java programming language was developed using both web services for searching films (title filtered) that contained outliers or missing data in order to look for the correct data and replacing them into the

corresponding row of the LODMatrix. Despite the web services were very useful to fix rows of the LODMatrix, there were still a large number of films with erroneous or missing data. For example, most of the films lacked the released year, so a java-based script was developed to extract years film's subjects.

At the end of the procedure for fixing the LODMatrix, some films still contained missing or erroneous data so the algorithms for recommendation using the LODMatrix had to support this kind of dataset (with missing and erroneous data). Additionally, the LODMatrix was the base for creating other datasets for further research, for example a binary dataset of films with their attributes as columns where each row is a film and the values for each column is 1 if the film contains such attribute and 0 otherwise. The description of these datasets as well as the implementation of the java program for creating the LODMatrix and its derived datasets is described in Appendix D.2

5.1.2 Query Controller

This component allows the recommendation algorithms to execute queries to access to the LODMatrix dataset. Therefore, this component contains file readers for accessing to different formats of the LODMatrix as required for example CSV and ARFF for the LODMatrix stored. Additionally, The lodmatrixdb was stored in a MySQL server, so the Query Controller component also contains mechanisms for executing SQL queries, in this case a JDBC controller and the DAO approach for accessing to relational database.

5.2 Recommender System Management layer

This layer contains the recommendation algorithms and the RS controller which controls the execution of these algorithms as well as the compositions or structures developed to test different behaviors of the recommendation algorithms.

5.2.1 Generation component

In this implementation the generation component contains the following sub-components: *1) Clustering:* groups films based on the similarity of the attributes that each film contains in the LODMatrix; 2) *Classification:* is used to assign new films, that were not included into the LODMatrix, to a cluster based on the results of the clustering component, i.e. a new film is assigned to a cluster based on the films that belong to the LODMatrix previously clustered.

The execution of the generation component can be divided in training phase and resource generation phase.

- *Training phase:* in this phase the clustering and classification algorithms are trained. The clustering component is trained to generate a clustering model to obtain a set of clusters where films are similar among them. Then, the classification algorithm is trained with the clustering data to obtain a classification model. The clustering data, is represented as a new attribute named *cluster* for the lodmatrix. This attribute is used for the classification model to infer the cluster of new films, i.e., films that are not stored in the lodmatrix. In this way, it is not necessary to train the clustering algorithm each time a new film is added to the lodmatrix, instead the classification model is used. Figure 5.2 represents the steps required for training the clustering and classification algorithms.

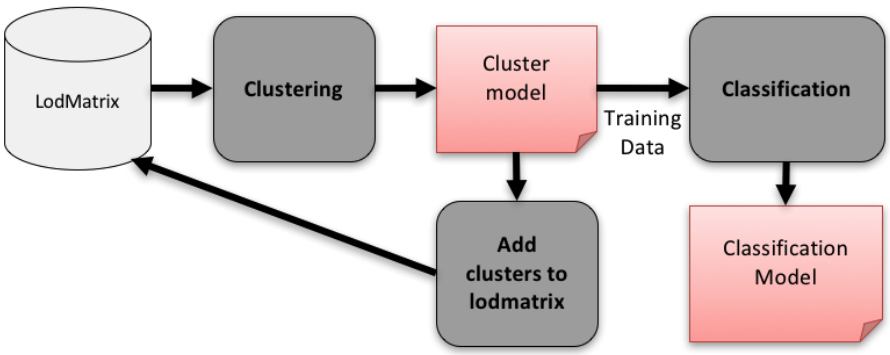


Figure 5.2. Training steps for the generation component

- *Resource generation:* in this phase an initial resource is received as query. Then, films that belong to the same cluster as the query film are retrieved. In case that the query film is not part of the lodmatrix (a new query) then the classification algorithm infers the cluster of the query film to extract the films that belong to the same cluster. This set of films of the cluster selected are then retrieved as response of the generation component. Figure 5.3, shows the process for generating candidate resources.

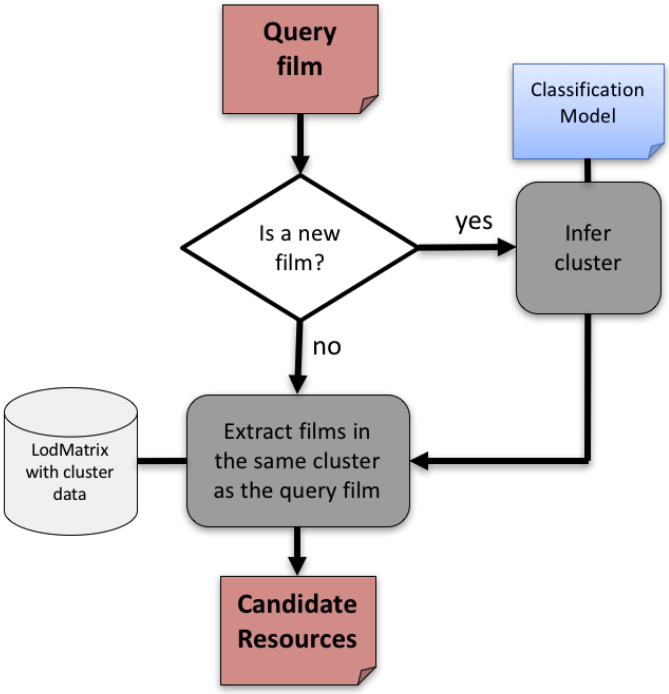


Figure 5.3. Steps for generate candidate resources

Clustering

Clustering is a technique of data mining which have been applied in a wide range of problems such as pattern recognition, image processing, knowledge discovery, and recommender systems. Clustering is concerned with grouping items together that are similar to each other and dissimilar to items belonging to other clusters. The similarity between items is based on a measure of the distance between them. In other words, items within one cluster are more similar among them that to any other items from the remaining clusters.

Recommendations based on clustering algorithms have been the subject of research in RS, but it has not been widely studied yet [101]. Most of these clustering algorithms have been used in combination with memory-based collaborative filtering approaches, therefore, in contrast to the current implementation, they require user's rating to produce recommendations. Consequently, these approaches use algorithms that compute distances between user profiles to identify neighbors. An user to whom recommendations are generated can be compared to their neighbors instead of all data, which considerably reduces computation time.

In the current implementation, there is no user's rating data or user's viewing history because it is intended to be a knowledge-based RS that rely solely on the information extracted from linked data to produce recommendations. It makes the current implementation suitable for RS that suffer of the clod-start problem.

The clustering approach is applied in two phases: an off-line phase which builds a clustering data model on the LODMatrix and an online-phase, where further calculation are preformed only on this model. In this way, recommendation of films for a query film are ranked based only on those films belonging to the same cluster, which reduces the computation time. Therefore, the proposed method is effective with regard to time during the on-line phase.

The generation layer used the most popular technique for clustering: the k-means algorithm. Initially, the k-means algorithm selects k points to be centers of k potential clusters, these points are known as the centroids. A centroid is the point for which each attribute value is the average of the values of the corresponding attribute for all the items in the cluster. Then, the algorithm assigns each item of the dataset (films of the LODMatrix) one by one to the cluster which has the nearest centroid (less distance), i.e. assign the items to their nearest cluster. When all the items have been assigned the centroids are recalculated and the previous steps are repeated until the centroids no longer needed to be recalculated.

Despite of the k-means algorithm is low-time complexity and it produces quite good quality clusters, the number of clusters (K value) should be predefined before the execution of the algorithm. However, the selection of K is difficult and it is commonly subjective of the application domain where the clustering algorithm is used. In the current implementation the clustering algorithm is used to reduce the complexity and time of the RS based on linked data, therefore clusters with few items while keeping as much as possible their quality are desirable.

Accordingly, an initial step for selection of the K value was conducted. There are some strategies for determining the number of clusters, for example: the X-means algorithm iteratively executes the K-means algorithm varying the K value within a predefined range of values to reach a value of K which best scores the Bayesian Information Criterion (BIC)[102]; the cascade simple k-means algorithm, also executes iteratively the k-means algorithm but it uses the calisnki-harabasz criterion[103]; approaches based on genetic algorithm optimization combined with the Self Organizing Map (SOM)[104], based on Gap Statistic [105], among others.

These approaches are based on the optimization of the item distances to the centroids, however this optimization is not always the best solution for some problems, for example in RS it is desirable to generate more clusters with less number of items each one, with the aim of reducing the execution time. Therefore, in the current implementation of the generation layer, the K value was chosen empirically by executing the K-means algorithm in the range 2 - 100 on the LODMatrix dataset and calculating the following measurements: centroid distance, distribution, and density.

1. **Centroid Distance:** evaluates the performance of the clustering model based on the centroids.

- *Average distance within centroid:* it is the average distance between the centroid and all examples of a cluster. Lower values of this measure are preferred since it means that the clusters have items grouped with smaller distance with respect to their centroid. Figure 5.4 shows the results for the average centroid distance for values of K in the range 2 - 100. The graphic shown that best values were scored in the range 55 - 100, with the best one in 55.

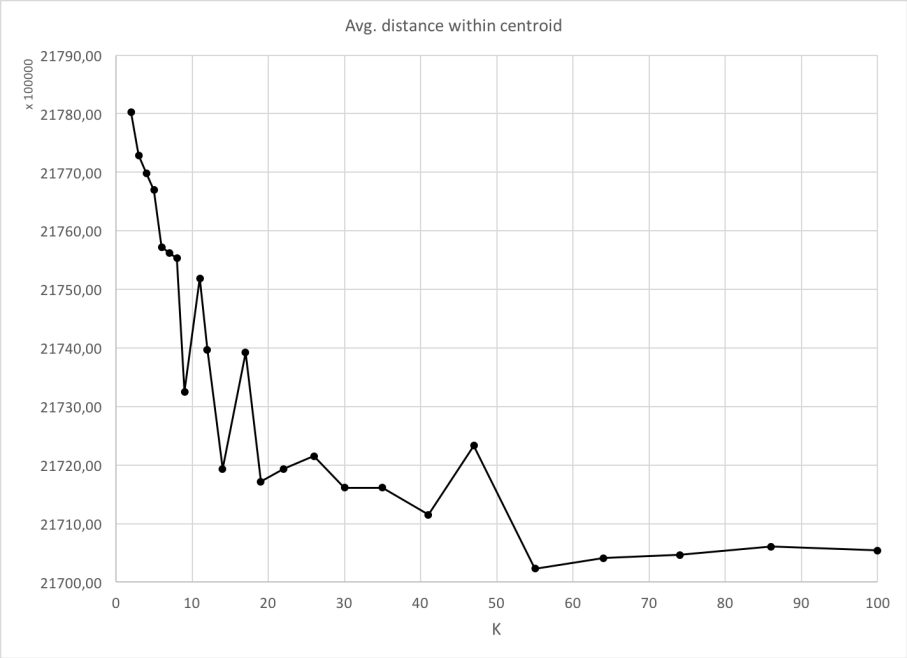


Figure 5.4. Average distance within centroid for K:{2-100}

2. **Distribution:** evaluates the performance of the model based on the distribution of items i.e. how well the items are distributed over the clusters. Two performance measures were computed:

   - *Sum of squares:* the number of items in each cluster is divided by the total number of items in the LODMatrix. This is squared and the values for each cluster are summed. For a situation where one cluster dominates and the others clusters are very small in comparison, this value will tend to 1. For the opposite situation, where the clusters have equal numbers of examples, the value tends to 1/K. Figure 5.5 shows two curves, the dotted one is the graphic for the 1/K values, and the plain one is the sum of squares. According, to the description for this measure the best value is 12 because it is the point where both curves are nearer each other.

However, values in the range 6 - 22 are also good points as they scored nearer points in both curves.

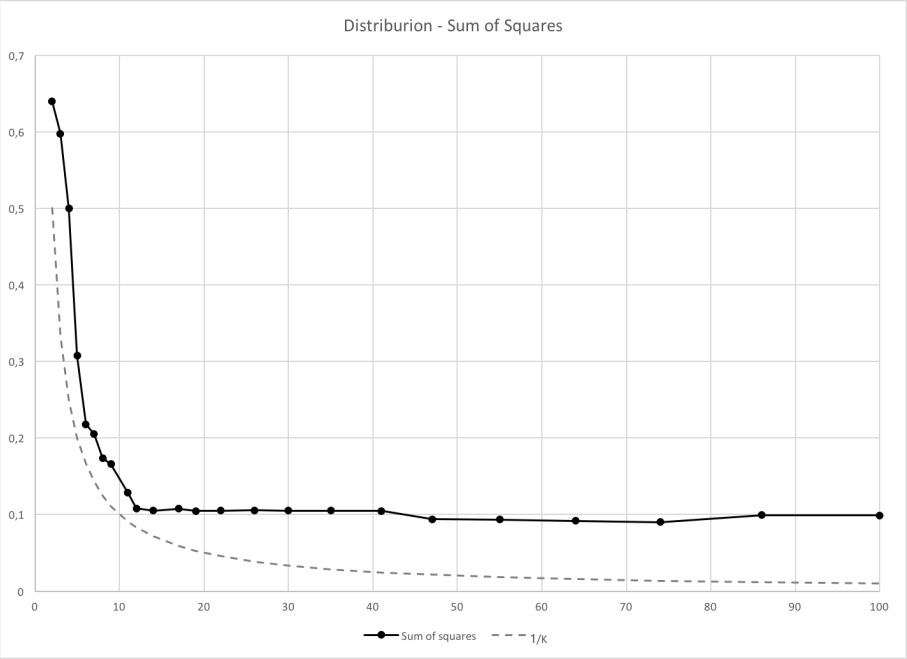


Figure 5.5. Sum of squares for K:[2-100]

- *Gini coefficient:* it is a measure of statistical dispersion, i.e., it measures the inequality among values of a frequency distribution. Low values of the Gini coefficient indicates a more equal distribution, then 0 is a complete equality, and 1 a complete inequality. Figure 5.6 shows that, according to the gini coefficient, most of the values in the range 2 - 200, scored values near to 1, i.e. the frequency distribution is mainly inequality. Anyway, the vest values were scored in the range 55 - 100.

3. **Density:** It is computed by averaging all distances between each pair of items of a cluster. Then the cluster density calculates the average distance between items in a cluster and multiplies this by the number of items minus 1. Commonly, the euclidean distance is used as the distance measure. The best density is the smallest value. Figure 5.7 shows that values in the range 40 - 100 scored the best values of density.

The experimentation for choosing the K value was performed with the software RapidMiner, the complete description as well as the process developed for the experimentation is presented in the Appendix D.4. The curves obtained for the experimentation shown different optimal values for the K value so it was not easy

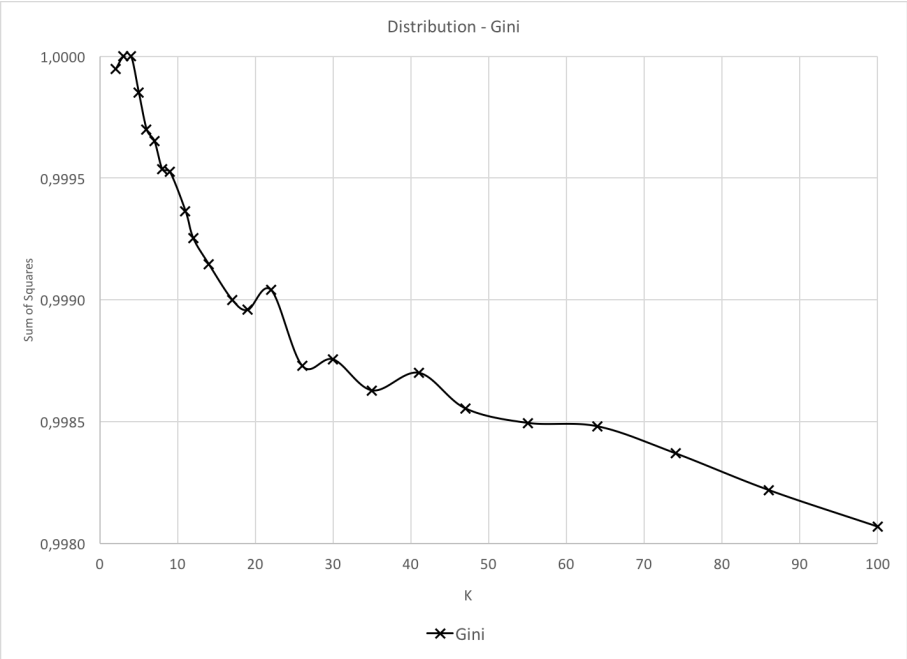


Figure 5.6. Gini coefficient for K:[2-100]

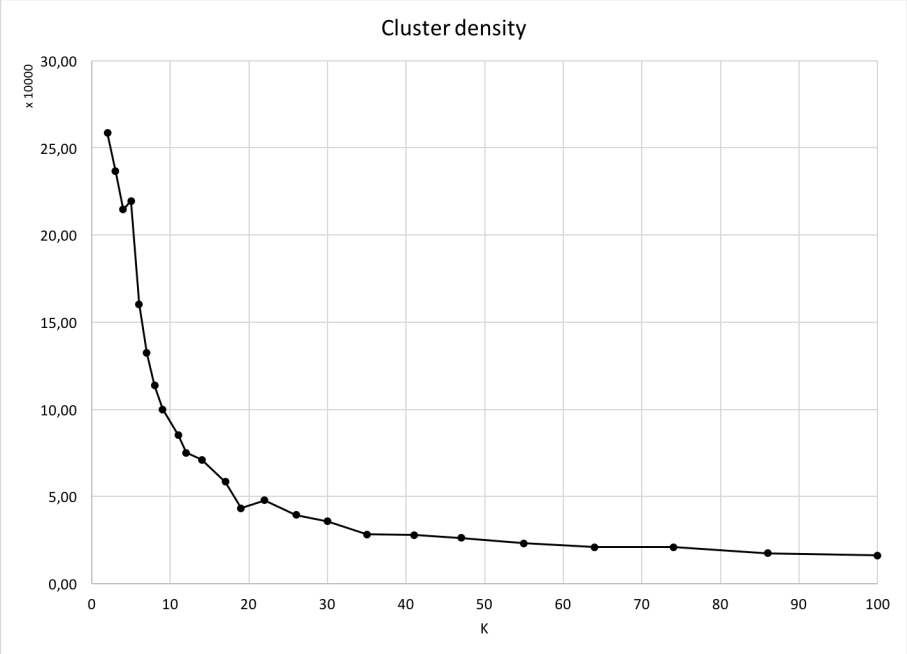


Figure 5.7. Density for K:[2-100]

to decide which one is the best. Therefore, the criteria for selecting the K value was to select a value that scored good values in all the curves. The value 55 scored pretty good values in all curves, including the curve for the sum of squares where this value was not to far from the optimal values scored in the range 6 - 22.

Classification

The classification phase was performed only for new films, i.e, when a new film is added, which was not present in DBpedia when films were retrieved to the knowledge base (LODMatrix). In such a case, clusters were used as classes and then various algorithms for classification were trained in order to infer the cluster (class) to which the new film belongs to.

The algorithms tested for the classification phase were the most commonly used for recommendation according to the State of the art presented in section 2.3.5: SVM, kNN, decision trees, random forest and naive bayes. The accuracy of the classification algorithms was evaluated using the confusion matrix approach.

Figure 5.8 shows the percentage of classification error for the algorithms selected. The classification error for each algorithm was computed using the confusion matrix which shows the relative number of misclassified items, i.e., the percentage of incorrect predictions of cluster produced by the classifier.

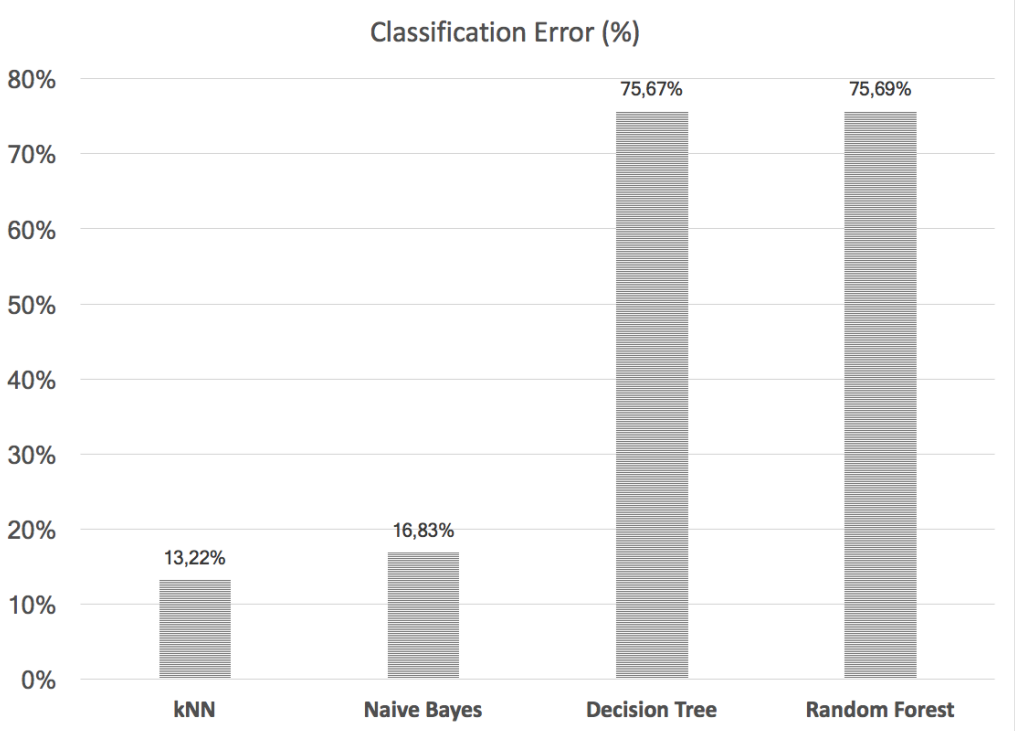


Figure 5.8. Classification error (%)

According to Figure 5.8, the best classifier for the LODMatrix is the kNN algorithm which scored the best value for the classification error (only 13.22%).

5.2.2 Ranking component

The *AlLied* framework in its current machine learning implementation includes (but is not limited) seven ranking algorithms. These functions are employed as measures of nearness between a query film and all the films belonging to the same set of clusters as the query. Taking into account that the LODMatrix is composed of nominal and numerical data, the ranking algorithms are based on similarity measures commonly used in machine learning for heterogeneous data.

Similarity Measures for Heterogeneous Data

Similarity measures are used to rank the films according to their similarity with the query film. The LODMatrix data are not suited for a specific measure so this study had to test different measures in order to determine which are better for ranking films in this study.

The similarity measures selected were the Mixed Euclidean distance and the Bregman distances because these measures are commonly studied in optimization, statistics, and machine learning[106].

- *Mixed euclidean distance:* is a heterogeneous distance measure capable to handle both numerical and nominal attributes, which is suitable for the LODMatrix. A formal description of this measure may be found at [107], where it is represented as equation 5.1:

$$MED = \sqrt{\sum_{A=1}^{m} d_A(x_A, y_A)^2} \tag{5.1}$$

  Where the distance between two values x_A and y_A is defined as equation 5.2:

$$d_A(x_A, y_A) = \begin{cases} overlap(x_A, y_A), & \text{if } A \text{ is categorical} \\ normdiff(x_A, y_A) & \text{if } A \text{ is numerical} \end{cases} \tag{5.2}$$

  Here, $overlap(x_A, y_A) = \begin{cases} 0, & x_A = y_A \\ 1 & otherwise \end{cases}$, and $normdiff = \frac{x_A - y_A}{max_A - min_A}$.

- *Bregman distances:* these measures are divergences (distances) that are also suitable for the LODMatrix, because they works with heterogeneous data and can be computed in time that scales quadratically with the rank of the input matrix[108]. Bregman divergences include various useful loss functions such as

generalized Bregman divergence, Itakura Saito, logarithmic loss, logistic Loss, squared euclidean, and squared loss. Formal descriptions of these measures may be found at [109].

The experimentation and selection of the most suitable measures for the LODMatrix is presented in chapter 6.2.

5.2.3 Grouping component

This layer was implemented in a similar way as the layer described for the graph-based implementations (chapter 4), in this case the URI of each film is extracted from the LODMatrix and categorized according to the hierarchical relationships of the Linked Data cloud. Algorithm 4 presented in the previous chapter is the implementation for the grouping layer.

This layer can be also implemented with a classifier algorithm, but in this case the layer may need a manual assignment of labels to each film. These labels may be the genres of the film and then perform a label classification for new films. In this way, films may be grouped based on the genres they belong to. However, this implementation does not included such grouping schema because it needed a manual operation to assign labels to each film for training the classifier. This is a future work, in which an automatic algorithm is being developed in order to extract genres from IMDB and label each film of the LODMatrix.

5.3 Presentation

This layer is similar to the presentation layer described in the graph-based implementation in chapter 4.

5.4 Summary

This chapter presented an implementation of the framework *AlLied* based on machine learning algorithms. The algorithms used in each layer are described as well as their selection criteria. The main drawback of using machine learning algorithms rather than graph-based algorithms for RS is that the data extraction step need to know a priory the application domain in order to obtain only relevant data of that domain. However, the application domain may be changed before the generation of the *lodmatrix*.

Chapter 6

Experimentation

This chapter regards the evaluation techniques used to study RS based on Linked Data. In this thesis the evaluation techniques were classified into two types: accuracy and computational performance. Accuracy evaluates recommendations according to their relevance, while computational complexity measures the execution time required to employed to generate and rank recommendations.

6.1 Evaluation for the graph-based algorithms

This study comparatively evaluated the prediction accuracy and the novelty of the resources recommended with ReDyAl with respect to three state-of-the-art recommendation algorithms based on relying exclusively on Linked Data structure (graph-based) to produce recommendations: *dbrec* [29], *HyProximity traversal* and *HyProximity hierarchical* [10]. 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?*

This part of the study is mainly focused in evaluating the novelty of the recommendations over the accuracy, then it relies on a user-based study. A user-based study measures novelty more precisely than an offline study because users rate items they already know, while discovering new unknown items that are relevant according to their personal criteria.

Although the *ReDyAl* algorithm is not bound to any particular dataset, this study used DBpedia because it is a general dataset that offers the possibility to evaluate the results in a number of scenarios and it is used by the related graph-based algorithms. DBpedia is one of the biggest datasets in the Web of Data and the most interlinked [91]. Furthermore, it is frequently updated and continuously grows.

6.1.1 Experimental setup

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. The domain of films was selected for the study due to facilitate the choice of the group of participants because no specific skills are required to express an opinion about films. The graph-based algorithms were compared within subjects [110] since each participant evaluated recommendations from different algorithms, as it is explained in the following.

This user study selected 45 query films out of the top 250 published in the website of IMDB[1]) as the set of possible initial queries for recommendation. Then a set of offline recommendations for each query and each algorithm was generated using the graph-based implementation of the *AlLied* framework. The algorithms involved in this experimentation are: hyProximity with hierarchical ranking, hyProximity with traversal ranking, dbrec with traversal ranking based on the LDSD distance, and ReDyAl which dynamically exploits both the traversal and hierarchical properties and a hybrid ranking.

Lists of the 20 more representative films for each query film were created. These lists were generated by merging the top 10 films in the recommendations that each algorithm generated for a given query film. Then, the lists of 20 films were delivered to the users so that they could evaluate the relevance of these recommendations according to each query film. For each item of the lists presented to the users two main questions were proposed in order to asses the accuracy and novelty of the recommendations:

(Q1) Did you already know this recommendation? Possible answers were: yes, yes but I haven't seen it (if it is a film) and no. (Q2) Is it related to the film 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. Additionally, other question was presented to receive a feedback from the users regarding to films that they considered relevant that did not appear into the presented lists.

Do you know other items, additionally to these 20 that you think are inherent to the initial film you've chosen?

The answers of the participants were collected through a website[2] collected in collaboration with other Ph.D student from Politecnico di Torino. Using this website the participants were able to choose an initial film from a list of 45 films (selected

[1] `http://www.imdb.com/chart/top`

[2] `http://natasha.polito.it/RSEvaluation/`

from the IMDB top 250 list). This choice ensured that a participant knew the selected film before the evaluation, however this also posed a limitation due to well-know films are also well-linked resources in DBpedia, so it was not possible to evaluate the algorithms on poorly linked films. The user interfaces for this evaluation are presented in Appendix E.

The initial page of the website presented the films for evaluation in a random order to avoid the participants to evaluate the same set of initial films (e.g. the first in the lists). When a participant selected an initial film the tool provided the corresponding list of recommendations with the questions mentioned above. The recommendations were presented in a randomized order. Each participant was able to evaluate recommendations from as many initial films as wanted, but the participant was required to answer the questions for all the recommendations, i.e. it was not possible to answer only a part of the questions for the initial film selected. As a result, the recommendations of the lists for 40 out of 45 initial films were evaluated by at least one participant and each film was evaluated by an average of 6.18 participants. The dataset with the initial films and the lists of recommendations is available online[3].

Each list of 20 recommendations was generated for each of the 45 initial films with each of the four graph-based algorithms. Then, as stated before, the recommendations generated by each algorithm were merged in a list of 20 recommendations to be shown to the participants. Accordingly, a list of 40 recommendations was generated by selecting the first 10 pre-computed recommendations for each algorithm and ascending ordered based on the similarity computed with each algorithms. After removing duplicated films, the final list was obtained considering only the first 20 recommendations of the merged list.

With regard to the questions stated at the beginning of this chapter, to answer RQ1, the Root Mean Squared Error (RMSE) [110] was computed, and to answer RQ2 the ratio between the number of evaluations was computed in which the recommended item was not known by the participants and the total number of evaluations. For the RMSE measure, scores given by the participants when answering the Q2 were considered as reference and were normalized in the interval [0, 1], and these scores were compared with the similarities computed by each algorithm (Each algorithm ranks its recommendations by using its semantic similarity function).

6.1.2 Results

The results of the evaluation are summarized in Figure 6.1, 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

[3] `http://natasha.polito.it/RSEvaluation/faces/resultsdownload.xhtml`

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 film which he has seen and which he liked may increase the trust of the user in the RS.

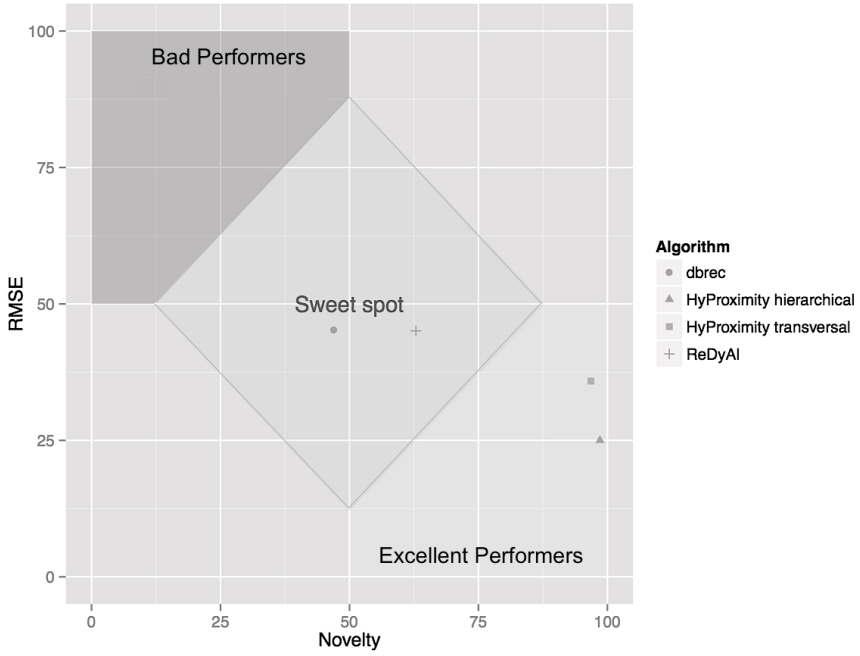


Figure 6.1. Prediction accuracy and novelty of the algorithms evaluated

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 keeping also good values for performance.

HyProximity generated recommendations based in both traversal and hierarchical algorithms, which only obtained few answers to Q2. In this regard, Table 6.1

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 specially meaningful for RQ1, since only the evaluations for which the answer to Q1 was was either "yes" or "yes but I haven't seen it (if it is a film)" were considered for computing the accuracy measures.

Algorithm	Yes	Yes, but I haven't seen it	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

Table 6.1. Percentage of answers for Q1 by algorithm

Furthermore, the Fleiss' kappa [111] measure was computed for assessing the agreement of the participants in 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 according to Landis and Koch [112] corresponds to a substantial agreement.

Figure 6.1 illustrates that ReDyAl and dbrec provides a good trade-off between prediction accuracy and novelty (sweet spot area), although ReDyAl performs better in novelty. HyProximity hierarchical and HyProximity traversal seem to be excellent performers since the RMSE is low and the novelty is high, but the RMSE was computed on few evaluations. An additional 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, further investigation is needed on poorly-linked resources, since the choice of the initial films focused on selecting well known films to make easier the evaluation from participants, but the related resources were well-linked. On poorly-linked resources ReDyAl and Hyproximity hierarchical are expected to keep good 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.

6.2 Evaluation for the machine learning algorithms

To determine the efficiency of the machine learning algorithms for recommending concepts, it was required to undergo an experimental evaluation to calculate the relevance of the results. The results obtained by each algorithm were compared with the judgements pronounced by human users as described in section 6.1.1; and also with the results of a gold-standard study.

6.2.1 User-study experimentation

The lists of relevant films for each query obtained from the user-study described in section 6.1.1 were also used to evaluate the results of the rankings generated in the machine learning implementation of the *AlLied* framework.

Experimental setup

In this experimental evaluation, the algorithms for were used to generate rankings of candidate resources in which lists of the first 10 candidate films more similarity with each query were considered. Accordingly, it was possible to assess the relevance of the results obtained in the execution of each algorithm, starting from the metrics widely used in the evaluation of information retrieval systems [113]: Precision (P), Recall (R), and *F-measure*.

Results

Figure 6.2 and table 6.2, show the values that each algorithm of the ranking layer for the machine learning implementation scored for precision, recall and F-measure.

	Mixed Euclidean	Generalizec Divergence	Itakura Saito	Logarithmi Loss	Logistic Loss	Squared Euclidean	Squared Loss
Precision	20,0%	9,1%	14,5%	13,6%	5,0%	14,1%	12,7%
Recall	22,0%	5,8%	12,2%	13,1%	3,9%	12,1%	12,0%
F-Measure	9,9%	3,5%	6,2%	6,4%	2,1%	6,4%	5,8%

Table 6.2. User study of relevance for rankers of the machine learning implementation

Mixed Euclidean distance scored the highest values (20.0%, 22.0%, and 9.9%) for the three measures followed by Itakura Saito (14.5%, 12.2%, and 6.2%) and Squared Euclidean (14.1%, 12.1%, and 6.4%) which barely achieved more than half of the values for precision, recall, and f-measure. Lowest results for the Bregman distances are due to they require the dataset to be specifically designed for them in order to obtain accurate results, while the Mixed Euclidean is a more generic measure which is simpler and therefore suitable for generic and heterogeneous datasets like the LODMatrix, which was not designed for a specific ranking algorithm.

6.2.2 Gold-standard experimentation

Taking into account that a user study may not be definitive to evaluate the accuracy of the recommender systems, in this section a gold-standard experimentations is presented. The gold standard experimentation, as opposed to the user study, is based

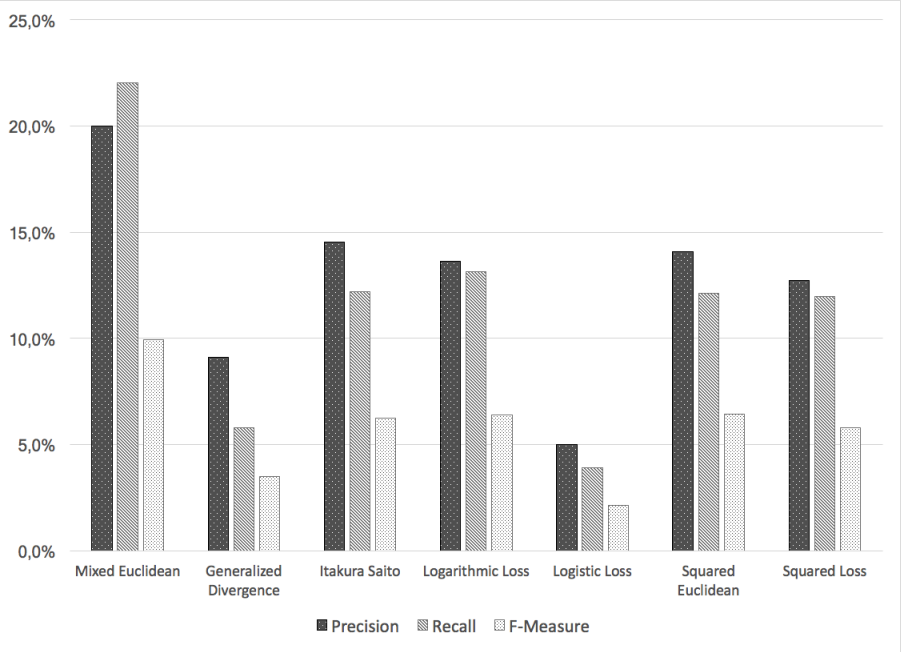


Figure 6.2. User study of relevance for rankers of the machine learning implementation

on the results of existing reliable systems which are considered as relevant in order to assess the accuracy of the proposed implementations of the *AlLied* framework.

Experimental setup

In this study, the *IMDb* system was chosen as a gold standard for high quality movies because it is one of the most reliable systems in evaluating movies and it has been widely used for evaluating other recommender systems [114].

The 10 top recommendations that *IMDb* posts on its website[4] for each query film were selected in order to create 40 lists of relevant films. These lists were used in order to asses the accuracy or the machine learning functions for ranking.

Results

Figure 6.3 and table 6.3 show the values that each algorithm of the ranking layer for the machine learning implementation scored for precision, recall and F-measure taking into account the top 10 films obtained from the IMDb for each query film.

[4] `http://www.imdb.com/`

	Mixed Euclidean	Generalizec Divergence	Itakura Saito	Logarithmi Loss	Logistic Loss	Squared Euclidean	Squared Loss
Precision	24,2%	2,5%	17,5%	20,0%	0,0%	10,8%	21,7%
Recall	20,1%	2,1%	14,6%	16,7%	0,0%	9,0%	18,1%
F-Measure	11,0%	1,1%	8,0%	9,1%	0,0%	4,9%	9,8%

Table 6.3. Gold-standard (IMDB) study of relevance for rankers of the machine learning implementation

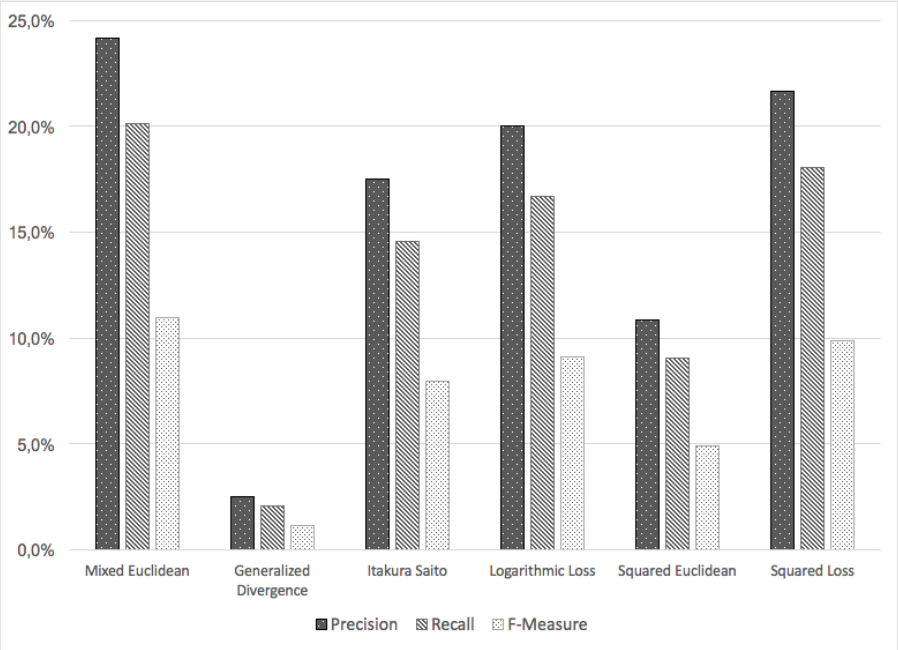


Figure 6.3. Gold-standard (IMDB) study of relevance for rankers of the machine learning implementation

Again, as demonstrated in section 6.2 the Mixed Euclidean distance scored the highest values (24.2%, 20.1%, and 11.0%) for all the measures. These results confirmed that Mixed Euclidean not only is the simplest measure for heterogeneous data but also the most accurate among the evaluated distances. However these values are not so good for the accuracy, this is mainly due to the relevant lists generated by users and the lists obtained from the gold-standard study only contained 10 main films for each query, and the number of films to be ranked, obtained in the generation layer, were about 3500 films by cluster. Therefore, it was difficult for a ranker algorithm to put all the relevant films on the top 10 of ranked list presented to the final user. As conclusion, more restrictive algorithms in the generation layer are necessary, which may be able to reduce the search space keeping a high value of recall, i.e., avoiding false negative items in the candidate films.

6.3 Comparative evaluation graph-based algorithms vs machine learning algorithms

The comparative evaluation between the graph-based vs machine learning algorithms compared the results obtained after ranking the films for each query as described in previous versions. In this study, only the most accurate algorithms were taken into account: ReDyal with Hybrid ranking for the graph-based algorithms and the Mixed Euclidean distance for the machine learning algorithms.

This comparative study was only based on the precision, recall and f-measure functions, because these measures were evaluated in both approaches the graph-based and the machine learning algorithms. Additionally, these measures were also considered in both studies the user and the gold-standard experimentation.

Firstly, the Figure 6.4, shows the results for accuracy obtained from the user study for *ReDyAl* with Hybrid ranking and the Mixed Euclidean function. In this study ReDyal scored highest values (43.2%, 35.7%, and 18.7%) for all the measures.

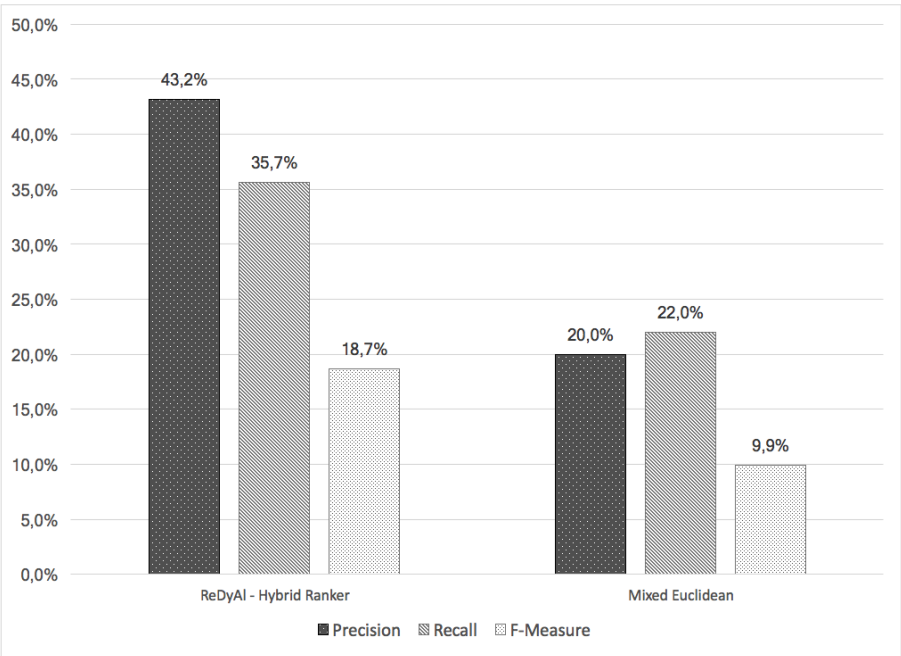


Figure 6.4. Comparative user study of relevance for graph-based and machine learning rankers

Secondly, similarly as the user study, the results for the gold-standard study presented in Figure 6.5 confirmed that ReDyAl outperformed the Mixed Euclidean function, which scored the highest values among the machine learning functions for ranking.

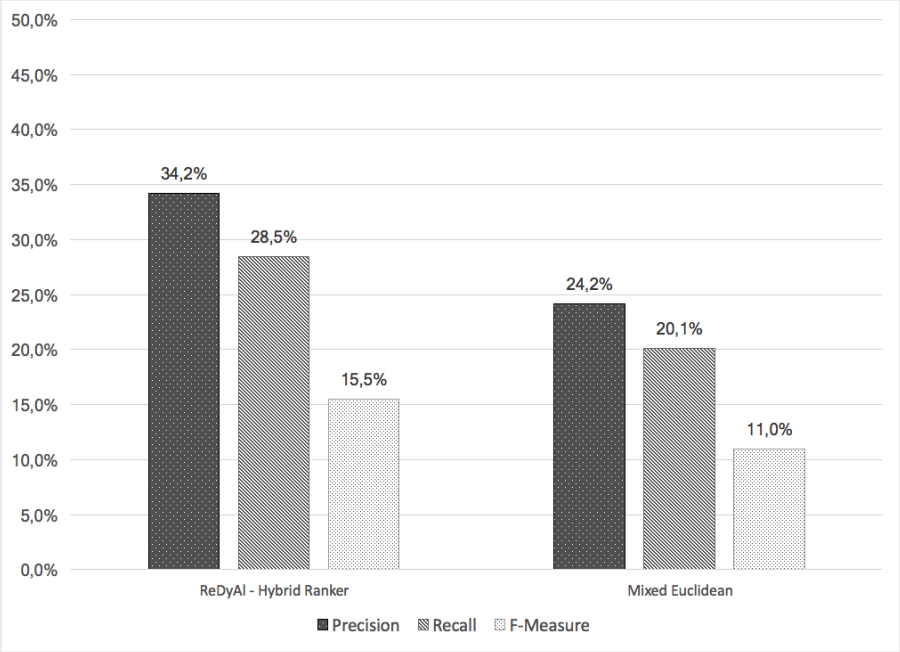


Figure 6.5. Comparative gold-standard (IMDB) study of relevance for graph-based and machine learning rankers

In both studies, ReDyAl outperformed the Mixed Euclidean function for all the measures. This result was expected as the ReDyAl with Hybrid Ranker is based on the graph structure of the Linked Data datasets, which makes it more suitable to take advantage of the semantic relationships between films. In other words, ReDyAl is an algorithm designed specifically to deal with LD datasets so it can work with the intrinsic relationships of the items represented through concepts of the web of data. Nevertheless, machine learning algorithms are being subject of future work, in order to adapt them to Linked Data datasets in a better way as they were used in this study to obtain more accurate results.

6.4 Evaluation of Performance

This experimentation was conducted in order to compute the execution time for the algorithms used in the recommendation process. The systematic review conducted in this thesis demonstrated that few works evaluated the performance of RS, which is a critical factor specially for applications that need responses with short timeouts. Therefore it is still an open issue considering that accessing to Linked Data datasets in most cases is time consuming and requires that researchers download dumps of the datasets to access them in local repositories.

Accordingly, this section presents the results of the evaluation of the performance for the generation layer and ranking layer, because the most critical tasks in the recommendation process, described in chapter 3 for the *AlLied* framework, are performed by these layers. This experimentation is divide in evaluation for graph-based algorithms and machine learning algorithms.

6.4.1 Performance for graph-based algorithms

This study evaluated the performance for the algorithms for generation and ranking layers. Table 6.4 shows the mean values for the execution time (in milliseconds) as well as the number of candidate films (resources) generated by the three algorithms for the generation layer in the graph-based implementation. The number of candidate films generated was different for each algorithm. The traversal generator shown to be the more restrictive as it extracted films that were only related by traversal links (direct or indirect links), while the hierarchical obtained the films that were related by hierarchical information which contains much more links than the traversal relationships. The ReDyAl algorithm generated an intermediate number of candidate resources as it uses dynamically the hierarchical and traversal relationships depending on the number of traversal links of the initial film (query).

With regard to the execution time, it was expected that even though the best relationship candidate resources-execution time was scored by the ReDyAl algorithm with a value of 3.7 resources per millisecond and the worse was scored by the Traversal generator with (0.28). This results shown, that generating candidate resources dynamically, not only allows to improve the accuracy and novelty but also the execution time.

Generator Algorithm	T (ms)	Candidate Films
ReDyAl	1911.3	7069.0
HierarchicalREC	5379.2	11513.1
Traversal REC	15637.5	4404.5

Table 6.4. Performance for generation layer algorithms

Table 6.5 shows the results of the performance for the ranking layer algorithms. The algorithms evaluated were the HybridSimpleRank, Traversal LDSD, and the Traversal HyProximity.

The ranker algorithms were tested with a fixed number of candidate resources, otherwise it is not possible to compare the algorithms. Though, this is not the real situation because the generation layer algorithms may generate different number of candidate resources depending on the initial film. The value of candidate resources selected for the experimentation was 23000 because it was the mean value among the generation algorithms presented in table 6.4.

This experimentation demonstrated that the faster ranking algorithm was the Traversal Ranker HyProximity and the lowest was the HybridSimpleRank. This was expected as the HybridSimpleRanker computes the similarity values based on both hierarchical and traversal links.

Ranking Algorithm	T(ms)
HybridSimpleRank	3545092
Traversal LDSD Rank	3255383
TraversalRankerHyProx	1454162

Table 6.5. Performance for ranking layer algorithms

The results for performance of the graph-based algorithms demonstrated that there is still a need for improving them because, even though the accuracy measures are good enough, the execution times are not suitable for applications where the final user wants a response time of only few milliseconds.

6.4.2 Performance for machine learning algorithms

This experimentation was conducted in order to compute the execution time for the algorithms used in the recommendation process. In this study the algorithms evaluated are: the algorithm k-means for clustering used in the generation layer; the algorithm kNN for classification (used with new films that are not part of the LODMatrix, that needed to be assigned to a cluster); and the ranking functions. Execution times were computed for two phases, because some machine learning algorithms require a previous phase for training know as offline execution and a second phase of testing known as online execution.

Offline execution

The offline execution is considered for those tasks of the machine learning that may be performed at different time than when the user is requiring recommendations, for example before the system is deployed for its first use, or in night hours when the system is rarely used. The two algorithms that needed a training phase were the k-means and the kNN and their execution time are presented in table 6.6. The most expensive task in the online execution was the clustering with the k-means, because it had to load the full LODMatrix to compute the similarity among all the films and locate them in clusters with similar films.

Offline execution	T(ms)
Training k-means	211440
Training kNN classifier	6000

Table 6.6. Offline execution

Online execution

The online execution is when the user is executing the recommender system, so the algorithms in this phase are: testing the KNN algorithm, only for those films that were not within the LODMatrix, so they needed to be located in one of the 55 clusters precomputed; searching for films in the same cluster as the query film; and the ranking films which sorts the films of a cluster according to their similarity with the query film.

Table 6.7 shows the execution times for the online tasks executed by the machine learning implementation of the *AlLied* framework. The most expensive task in the online execution was testing the KNN algorithm, however this task is only used when a new film is added to the LODMatrix as explained before. The second expensive task is the searching for films in the same cluster as the query film followed nearly by the ranking task. The time of the ranking task was computed as the mean among 100 execution of the ranking functions.

Online execution	T(ms)
Testing KNN	60000
Searching for films in the same cluster as the query	5000
Ranking	8000

Table 6.7. Online execution

6.5 Concluding Remarks

This chapter demonstrated that graph-based algorithms are more accurate than machine learning algorithms. This is because the former algorithms are specifically developed to deal with Linked Data datasets and they can take advantage of the intrinsic relationships among the items represented through resources of the web of data. Nevertheless, the execution time of the graph-based algorithms is still an open issue because they far exceed the execution time of the machine learning algorithms.

Machine learning algorithms allowed the framework to improve its performance because are designed to execute tasks for large amounts of data. Therefore, a logical evolution of RS based on Linked Data is the combination of graph-based algorithms with machine learning algorithms, in order to obtain accurate results in an execution time adapted to the needs of the end users.

However, research and experimentation is still needed in RS based on Linked Data, to explore more techniques from the vast amount of machine learning algorithms to determine which of them are more suitable to deal with Linked Data datasets.

6.6 Tools

The experiments described in this chapter were executed in a computer with this features:

- RAM Memmory: 32 GB
- Processor: Intel Core I7 2.6GHz

The software tools used were:

- Weka 3.8: this software contains a large amount of machine learning algorithms. It was used for the clustering phase specially the k-means algorithm. Additionally, the LOF algorithm for outliers detection was also executed with this software.
- R: this software was used for the tukey's algorithm for outliers detection as well for testing other algorithms in machine learning.
- RapidMiner: this software also contains a large amount of operations for data mining, including text processing, recommender systems, among others. This software was used for executing most of the experiments in the machine learning implementation. For example, the execution of the KNN algorithm for classification, the ranking functions, and also an implementation of the Weka's k-means algorithm.

6.7 Summary

This chapter presented the experimental setup as well as the results obtained to asses the performance and accuracy of the algorithms proposed for both implementations, the graph-based and the machine learning, for the *AlLied* framework. Results shown that even though graph-based algorithms were more accurate, they are more time-consuming than machine learning algorithms, which needs to be studied in future research to improve their results.

Chapter 7

Conclusions and Future Work

This thesis presented a study of Recommender Systems (RS) based on Linked Data. The foundation of this thesis is a Systematic Literature Review (SLR), which highlighted that there are still open issues in this research field such as: the need for creating local copies of the Linked Data datasets to reduce the runtime to produce recommendations; manual selection of a subset of resources belonging to a specific domain to create domain-dependent RS; and the cold-start problem for RS that require user information such as rating, user profile, and user history to produce recommendations.

Furthermore, the SLR proposed a classification of Linked Data based RS composed of four categories: *Linked Data driven RS* relies solely on Linked Data knowledge to perform their tasks; *hybrid RS* uses Linked Data but also other techniques; *representation only RS* does not provide Linked Data-based recommendations but it uses Linked Data for representing data based on RDF; and finally *exploratory search systems* that are not RS but may help users to find concepts or topics, these systems have some similar features to RS especially in the use of Linked Data.

Accordingly, a framework for deploying and executing recommendation algorithms based on Linked Data as their knowledge base dubbed *AlLied* was proposed. This framework facilitated the prototyping and benchmarking of different algorithms, as they were deployed in the same environment and the generated recommendations were aligned. Therefore, it enabled to measure and compare the accuracy and performance of the algorithms, in order to determine which algorithms are the best recommenders of resources from the Web of Data when focusing on a specific application or domain.

The current implementation of the *AlLied* framework includes two variants: the first one with graph-based algorithms that take advantage of the intrinsic semantic structure of the Linked Data datasets, and the second one composed of machine learning algorithms. Both implementations contains algorithms already developed. These algorithms were studied experimentally with *AlLied*: the graph-based implementation scored best values of accuracy as its algorithms are specifically designed

to deal with Linked Data datasets, however they shown a poor performance due to excessive execution times. On the contrary, the machine learning implementation scored lowest values of accuracy because its algorithms are generic and can be used in a variety of applications, nevertheless as they are designed to execute tasks for large amounts of data (e.g., tasks commonly used in BigData), they offer execution times that have been optimized for real applications, which require responses in only few milliseconds.

An algorithm named *ReDyAl* was developed and deployed into the *AlLied* framework. *ReDyAl* is a hybrid graph-based algorithm, that dynamically integrates both the traversal and hierarchical approaches for discovering resources. It was designed based on the analysis of state-of-the-art recommendation algorithms and it was also deployed within the *AlLied* framework.

The algorithms implemented within *AlLied*, including *ReDyAl* were evaluated and compared by conducting a user experimentation and a gold standard study. The results of these studies demonstrated that *ReDyAl* improved in the novelty of the results discovered, although the accuracy of the algorithm is not the highest (due to its inherent complexity). ReDyAl is not bounded to any particular domain, but the study focused on films because in this domain a quite large amount of data is available on DBpedia and finding participants is easy, since no specific skills are required. In this case, *AlLied* is useful to repeat the study in any other domain.

Additionally, a complete list of the research papers published during the PhD are presented in the Appendix F.

7.1 Conclusions

The main conclusions derived from the study described in this thesis are:

- The *AlLied* framework is the first an architecture for Linked Data based RS, which divides the recommendation process in meaningful phases that are embodied by layers. In other words, each layer represents one task of the recommendation process that may be executed by various algorithms. In this way, the *AlLied* framework allows the developers to develop and evaluate different configurations (combinations) of algorithms for each layer to develop novel RS based on Linked Data suitable to users' requirements, applications, domains and contexts.

- This layered architecture of the *AlLied* framework is also useful towards the reproducibility of the results for the research community of RS because the recommendation process is divided in different algorithms (for each layer), which may be tuned to improve the accuracy and performance of the overall RS for a specific application. This is specially important taking into account that the difficulty to reproduce results is widely agreed in the community[115].

- The *ReDyal* algorithm was developed based on the experimentation conducted on graph-based state of the art algorithms. This study demonstrated that some of these algorithms are more suitable to generate candidate resources under certain conditions of the initial resource. For example, if the initial resource contains a minimum number of links (properties) to other resources it should be desirable to execute the traversal algorithm that is specialized on finding related resources through the traversal links. The *ReDyal* automatically chooses the best algorithm to recommend resources based on the links of the initial resource.

- This study has demonstrated that The *ReDyal* is able to generate novel recommendations. This is useful when users do not want to receive recommendations about items they already know or have previously consumed. Additionally, recommending very popular items, which can be easily discovered may not be enough. For this reason, a high novelty score in the recommendations is important to propose items that are interesting and unexpected.

- Linked data based RS are not only capable to generate items of a specific type but also items of different types. For example, although most of the results obtained by the RS implemented in this thesis were films, other results belonged to different types like producers, actors, cities, etc. This feature may be an advantage in the case of RS that require the use of heterogeneous information such as tourism in which recommended items may not be only points of interest such as museums, but also important personalities living in the city, typical foods, among others.

- Linked data based RS are useful to present explanations of the recommendations, because of the graph structure of the datasets in which the items are interlinked. In this case, following the links of the graph to which the recommended items belong is enough.

- Graph-based algorithms are specifically developed to deal with Linked Data datasets and they can take advantage of the intrinsic relationships among the items represented through resources of the web of data. Nevertheless, the execution time of the graph-based algorithms is still an open issue because they far exceed the execution time of the machine learning algorithms.

- Machine Learning algorithms are also suitable for recommendations based on Linked Data, they provide a large set of functions useful to deal with large amounts of data. This is important for the *AlLied* framework because it is planned to work with not only one dataset as presented in this thesis, but also with multiple and heterogeneous dataset.

- Machine Learning implementation contains a classification algorithm which allows the RS to work with unknown films (films that are not within the lodmatrix). To do this, the classification algorithm infers the clustering of the query film based on the classification model built with the training data obtained from the results of the clustering algorithms.

- The main drawback of using machine learning algorithms rather than graph-based algorithms for RS is that most of them need to know a priory the application domain in order to obtain only relevant data of that domain. However, as described in chapter 5, the application domain may be changed before the execution of the SPARQL queries used for creating the *lodmatrix*.

- Although the state of the art demonstrated that there are various RS that use machine learning algorithms, most of them require user profile information (e.g., ratings, user history, user information) to produce recommendations. However, when a new user or item is added to the RS, it produces the problem known as cold-start. This thesis has presented the use of these algorithms in order to produce recommendations without user profile information. The *AlLied* framework works solely using the semantic knowledge extracted from the item features and their relationships with resources of the Linked Data.

- A logical evolution of RS based on Linked Data is the combination of graph-based algorithms with machine learning algorithms, in order to obtain accurate results in an execution time adapted to the needs of the end users. However, research and experimentation is still needed in RS based on Linked Data, to explore more techniques from the vast amount of machine learning algorithms to determine which of them are more suitable to deal with Linked Data datasets.

- Although, this study is focused only on RS that does not require user information, the knowledge of Linked Data datasets should not be limited to exploit relationships among items. Linked Data may be also useful to enrich items and users in order to generate implicit knowledge about them and their relationships. In this way, RS would be able to produce personalized recommendations relying on the derived implicit knowledge.

- In the recent years, the interest for RS based on Linked Data has increased. However, their use and performance are still subject of research to obtain resources with a good degree of accuracy while keeping low-execution times. Therefore, this research project encourages other scientists to study this interesting type of RS using the *AlLied* framework to create compositions (combinations) of recommendation algorithms and to determine which algorithms are more suitable in a desired situation.

7.2 Proof of concept / use cases

This section presents how the *AlLied* framework as well as the *ReDyAl* algorithm have been applied to various scenarios to demonstrate their feasibility within the context of real applications.

- The *ReDyAl* has been integrated into a mobile application developed in collaboration with Telecom Italia. This application recommends movies based on DBpedia: when the user enters the title of a movie, the application provides the Wikipedia categories to which the initial movie is related to. In this way, the user may focus on a specific scope and can receive recommendations of related resources for any category. In addition, it is possible to view any recommendation to obtain additional information. *ReDyAl* can provide cross-domain recommendations because it is independent on the domain and is applied on DBpedia, which is a general dataset. Thus, the recommended resources can be movies but also other relevant entities such as actors, directors, places of recording, books on which the movie is inspired, etc. Other advantages of using DBpedia as dataset are the high number of resources that it represents, the variety of domains addressed and the continuous update and growth, since it is extracted from Wikipedia. The recommender service was developed in Java, while the client is a mobile application developed for the Android operating system. The mobile application is going to be published on Google Play, but the Android Package (APK) of the first version is already available on the Web[1]. This work was published in a conference paper entitled *"ReDyAl: A Dynamic Recommendation Algorithm based on Linked Data"*[6], which was presented into the *"3rd Workshop on New Trends in Content-based Recommender Systems - CBRecSys within the ACM RecSys 2016"*, which is one of the most important conferences about RS.

- The hierarchical generator presented in this thesis was integrated into a framework for Multimodal Search of Business processes. The hierarchical generator extracted categories for Business Process activities to create a hierarchical index for searching Business Processes on a repository. The search process on a categorized repository presented a significant reduction on the execution time because the search space is not the whole repository but only those BP that belong to a similar set of categories. This work was published in the *"Knowledge-based Systems"* journal [116].

- The *ReDyAl* algorithm was used to improve a software named *TellMeFirst*.

[1] `https://www.dropbox.com/sh/0q8d2mcbko9e2oj/AAASh-YHGz0MmG_Z8hH6mfWOa?dl=0`

TellMeFirst is a software for the classification and enrichment of textual documents written in English and Italian. It was adopted by a telecommunications operator to add value to its mobile services: FriendTV and SOCIETY. This work was performed in collaboration with the main Telecommunications Operator in Italy, and in this way to put into direct contact common users of mobile services with the Web of Data. Then *ReDyAl* was integrated to improve the functionalities of TellMeFirst by suggesting similar resources related to those originally extracted by the TellMeFirst's semantic annotator. This improvement has enabled a whole new scenario of multi-domain recommendations. Additionally, many efforts has been made to adapt the operation of TellMeFirst with multiple knowledge bases (and not just DBpedia). As a result of this collaboration with Telecom Italia a joint research paper was published into the *"IEEE ITPro Magazine"* entitled *"Semantic Annotation and Classification in Practice"*[117].

- The *AlLied* framework was used to develop a platform for exploiting the knowledge in the Web of Data in the context of Smart Spaces. Smart Spaces are any real or virtual location equipped with passive and active artifacts. These artifacts can be any kind of sensors and actuators, mobile devices, software agents, autonomous vehicles, management systems, and also human beings. Due to the large amount of data that exist in the Web of Data, it was possible to find related structured information about many of the components (artifacts) that are part of a specific Smart Space. At the same time, it is possible that some components of the Smart Space (e.g. users' devices) enrich the existing information about other components by generating semantic annotated user-generated content (UGC). To do this, the RS was integrated with an enabler of the FI- WARE EU project [2].This platform obtains information about Smart Spaces components to determine the mode of interaction between these components, i.e. a component can make decisions about how to act based on the information that it receives about the other components. Such interactions can be influenced through the use of a LD-driven recommender created with the AlLied framework. Some Smart Spaces components such as the users' device can generate content and enrich the information of another Smart Spaces components by means of a semantic annotation process. Finally, an eTourism use case which was modeled and developed on top of it, in conjunction with a mobile operator. This use case was presented in the *"The 7th conference on Internet of Things and Smart Spaces ruSMART 2014"*[118].

[2]FI-WARE is a project funded by the European Comission. More information at `http://www.fi-ware.org`

7.3 Future Work

- Future work includes a deep study of more types of algorithms such as evolutionary computation, automated planning, among others in order to study the relevance under different domains and improving the performance and accuracy of *ReDyAl* while maintaining its novelty. In this way, *ReDyAl* can discover resources faster and can be usable for real-time applications.

- Linked Data can be also used to explain recommendations since they encode semantic information in a graph structure. This is specially useful when unknown items are proposed, in this case the system should assist the user in the decision process, both to justify the suggestion and to provide additional information to understand the quality of the recommended item. This may increase the transparency of the system and the user's trust and satisfaction.

- Currently, new algorithms are being implemented into the *AlLied* framework to increase the available set of techniques for each layer. However, a detailed study of the accuracy and performance measures for each layer of the framework is needed. This study, will improve the selection of the algorithms for each layer.

- A limitation that needs to be addressed in the future works consists in the lack of personalization. The algorithms currently implemented within *AlLied* rely exclusively on Linked Data to generate recommendations, and do not provide personalized recommendations. However, they can be effectively applied to situations such as cold start. The algorithms currently implemented are suitable to this kind of situations. Furthermore, the framework is designed to be extended, thus collaborative filtering algorithms can be added in future versions in order to personalize the recommendations.

- There are still open problems which require further research. For example, discovering latent relationships among items and users could enable diversified recommendations. Diversity is a popular topic in content-based recommender systems, which usually suffer from overspecialization.

- Other issue that may be addressed in future versions of the *AlLied* is mining microblogging data and text reviews. Opinion mining and sentiment analysis techniques can support recommendation methods that take into account 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.

- A closely related research area is the 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 semantics richness of the data for successful exploratory search.

7.4 Summary

This chapter summarized the main contributions of this thesis, described the conclusions obtained from this study, presented the application scenarios where the proposed framework was used, and described the future works to complement this study.

Bibliography

[1] D. Wood, M. Lanthaler, and R. Cyganiak, "RDF 1.1 concepts and abstract syntax," W3C recommendation, W3C, Feb. 2014. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

[2] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook," in *Recommender Systems Handbook*, pp. 1–35, Springer, 2011.

[3] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data - the story so far," *Int. J. Semantic Web Inf. Syst.*, vol. 5, no. 3, pp. 1–22, 2009.

[4] C. Figueroa, I. Vagliano, O. Rodríguez-Rocha, and M. Morisio, "A systematic literature review of linked data-based recommender systems," *Concurrency and Computations: Practice and Experience*, 2015. In press.

[5] C. Figueroa, I. Vagliano, O. Rodríguez-Rocha, M. Torchiano, C. Faron-Zucker, J.-C. Corrales, and M. Morisio, "Allied: A Framework for Executing Linked Data-based Recommendation Algorithms," *International Journal on Semantic Web and Information Systems*, vol. 13, no. (3) 2017, in press.

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

[7] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender systems handbook*, pp. 73–105, Springer, 2011.

[8] A. Felfernig, M. Jeran, G. Ninaus, F. Reinfrank, and S. Reiterer, "Toward the Next Generation of Recommender Systems: Applications and Research Challenges," in *Multimedia Services in Intelligent Environments*, ch. Smart Inno, pp. 81 – 98, Springer, 2013.

[9] D. Dell'Aglio, I. Celino, and D. Cerizza, "Anatomy of a semantic web-enabled knowledge-based recommender system," in *CEUR Workshop Proceedings*, vol. 667, pp. 115–130, 2010.

[10] D. Damljanovic, M. Stankovic, and P. Laublet, "Linked data-based concept

recommendation: Comparison of different methods in open innovation scenario," in *The Semantic Web: Research and Applications* (E. Simperl, P. Cimiano, A. Polleres, O. Corcho, and V. Presutti, eds.), no. 7295 in Lecture Notes in Computer Science, pp. 24–38, Springer Berlin Heidelberg, Jan. 2012.

[11] T. Berners-Lee, "Linked Data - Design Issues," 2006.

[12] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, "A literature review and classification of recommender systems research," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10059–10072, 2012.

[13] B. Kitchenham, "Procedures for Performing Systematic Reviews," tech. rep., Keele University, Eversleigh, Australia, 2004.

[14] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," tech. rep., University of Durham, Durham, UK, 2007.

[15] B. Heitmann, R. Cyganiak, C. Hayes, and S. Decker, "An empirically grounded conceptual architecture for applications on the web of data," *Trans. Sys. Man Cyber Part C*, vol. 42, pp. 51–60, Jan. 2012.

[16] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, pp. 34–43, May 2001.

[17] I. Jacobs and N. Walsh, "Architecture of the world wide web, volume one," W3C recommendation, W3C, Dec. 2004. http://www.w3.org/TR/2004/REC-webarch-20041215/.

[18] A. Passant, "Measuring semantic distance on linking data and using it for resources recommendations," in *AAAI Spring Symposium - Technical Report*, vol. SS-10-07, pp. 93–98, 2010.

[19] R. Krummenacher, B. Norton, and A. Marte, "Towards linked open services and processes," in *Future Internet - FIS 2010*, vol. 6369 of *Lecture Notes in Computer Science*, pp. 68–77, Springer Berlin Heidelberg, 2010.

[20] S. Harris and A. Seaborne, "SPARQL 1.1 query language," W3C recommendation, W3C, Mar. 2013. http://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[21] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734–749, 2005.

[22] B. Heitmann and C. Hayes, "Using linked data to build open, collaborative recommender systems," in *AAAI Spring Symposium - Technical Report*, vol. SS-10-07, pp. 76–81, 2010.

[23] L. Candillier, K. Jack, F. Fessant, and F. Meyer, "State-of-the-art recommender systems," *Collaborative and Social Information Retrieval and Access-Techniques for Improved User Modeling*, pp. 1–22, 2009.

[24] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, no. 0, pp. 109 – 132, 2013.

[25] R. Mirizzi and T. Di Noia, "From exploratory search to web search and back," in *Proceedings of the 3rd Workshop on Ph.D. Students in Information and Knowledge Management*, PIKM '10, (New York, NY, USA), pp. 39–46, ACM, 2010.

[26] M. McShane, S. Nirenburg, and S. Beale, "NLP with reasoning and for reasoning," in *Ontology and the Lexicon: A Natural Language Processing Perspective* (C.-r. Huang, N. Calzolari, A. Gangemi, A. Lenci, A. Oltramari, and L. Prevot, eds.), ch. Ontology„ pp. 98–121, Cambridge: Cambridge University Press, 2010.

[27] J. Urbani, J. Maassen, N. Drost, F. Seinstra, and H. Bal, "Scalable rdf data compression with mapreduce," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 1, pp. 24–39, 2013.

[28] M. Welling, *A First Encounter with Machine Learning*. Donald Bren School of Information and Computer Science - University of California Irvine, 2011.

[29] A. Passant, *dbrec — Music Recommendations Using DBpedia*, pp. 209–224. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[30] R. Yang, W. Hu, and Y. Qu, "Using Semantic Technology to Improve Recommender Systems Based on Slope One," *Springer Proceedings in Complexity*, 2013.

[31] K. Kitaya, H.-H. Huang, and K. Kawagoe, "Music curator recommendations using linked data," in *2012 Second International Conference on Innovative Computing Technology (INTECH)*, pp. 337–339, Sept. 2012.

[32] C. Musto, P. Basile, P. Lops, M. De Gemmis, and G. Semeraro, "Linked open data-enabled strategies for top-n recommendations," in *CEUR Workshop Proceedings*, vol. 1245, pp. 49–55, 2014.

[33] P. Nguyen, P. Tomeo, T. Di Noia, and E. Di Sciascio, "An evaluation of simrank and personalized pagerank to build a recommender system for the web of data," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 1477–1482, ACM, 2015.

[34] R. Blanco, B. B. Cambazoglu, P. Mika, and N. Torzec, "Entity recommendations in web search," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8219 LNCS, pp. 33–48, 2013.

[35] H. G. Ko, E. Kim, I. Y. Ko, and D. Chang, "Semantically-based recommendation by using semantic clusters of users' viewing history," in *2014 International Conference on Big Data and Smart Computing, BIGCOMP 2014*, pp. 83–87, 2014.

[36] S. Baumann, R. Schirru, and B. Streit, "Towards a storytelling approach for

novel artist recommendations," in *Adaptive Multimedia Retrieval. Context, Exploration, and Fusion* (M. Detyniecki, P. Knees, A. NÃ¼rnberger, M. Schedl, and S. Stober, eds.), no. 6817 in Lecture Notes in Computer Science, pp. 1–15, Springer Berlin Heidelberg, Jan. 2011.

[37] H. Khrouf and R. Troncy, "Hybrid event recommendation using linked data and user diversity," in *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, (New York, NY, USA), pp. 185–192, ACM, 2013.

[38] V. C. Ostuni, G. Gentile, T. Di Noia, R. Mirizzi, D. Romito, and E. Di Sciascio, "Mobile movie recommendations with linked data," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8127 LNCS, pp. 400–415, 2013.

[39] A. Hajra, A. Latif, and K. Tochtermann, "Retrieving and ranking scientific publications from linked open data repositories," in *Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business*, p. 29, ACM, 2014.

[40] N. Marie, F. Gandon, M. RibiÃ¨re, and F. Rodio, "Discovery hub: On-the-fly linked data exploratory search," in *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, (New York, NY, USA), pp. 17–24, ACM, 2013.

[41] Y. Yu, J. Kim, K. Shin, and G. S. Jo, "Recommendation system using location-based ontology on wireless internet: An example of collective intelligence by using mashup applications," *Expert Systems with Applications*, vol. 36, pp. 11675–11681, Nov. 2009.

[42] I. Cantador, P. Castells, and A. Bellogín, "An enhanced semantic layer for hybrid recommender systems: Application to news recommendation," *Int. J. Semant. Web Inf. Syst.*, vol. 7, pp. 44–78, Jan. 2011.

[43] M. Kaminskas, I. Fernández-Tobías, F. Ricci, and I. Cantador, "Knowledge-based music retrieval for places of interest," *Second international ACM workshop on Music information retrieval with user-centered and multimodal strategies - MIRUM '12*, p. 19, 2012.

[44] I. Fernández-Tobías, I. Cantador, M. Kaminskas, and F. Ricci, "A generic semantic-based framework for cross-domain recommendation," *2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems - HetRec '11*, pp. 25–32, 2011.

[45] S. K. Cheekula, P. Kapanipathi, D. Doran, and P. Jain, "Entity Recommendations Using Hierarchical Knowledge Bases," *Proceedings of the 4th workshop on Knowledge Discovery and Data Mining Meets Linked Open Data at ESWC2015*, pp. 1–12, 2015.

[46] J. Chicaiza, N. Piedra, J. López-Vargas, and Tovar-Edmundo, "Domain Categorization of Open Educational Resources Based on Linked Data," *Knowledge*

Engineering and the Semantic Web, pp. 15–28, 2014.

[47] I. Cantador, I. Konstas, and J. M. Jose, "Categorising social tags to improve folksonomy-based recommendations," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 9, no. 1, pp. 1 – 15, 2011.

[48] L. Strobin and A. Niewiadomski, "Recommendations and object discovery in graph databases using path semantic analysis," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8467 LNAI, pp. 793–804, 2014.

[49] P. Ristoski, E. L. Mencía, and H. Paulheim, "A hybrid multi-strategy recommender system using Linked Open Data," *Communications in Computer and Information Science*, vol. 475, pp. 150–156, 2014.

[50] J.-w. Ahn and X. Amatriain, "Towards fully distributed and privacy-preserving recommendations via expert collaborative filtering and RESTful linked data," in *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, vol. 1, pp. 66–73, Aug. 2010.

[51] V. C. Ostuni, T. Di Noia, E. Di Sciascio, and R. Mirizzi, "Top-n recommendations from implicit feedback leveraging linked open data," in *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, (New York, NY, USA), pp. 85–92, ACM, 2013.

[52] A. Moreno, C. Ariza-Porras, P. Lago, C. L. Jiménez-Guarín, H. Castro, and M. Riveill, "Hybrid model rating prediction with linked open data for recommender systems," in *SemWebEval 2014 at ESWC 2014*, vol. 475, pp. 193–198, 2014.

[53] F. Narducci, C. Musto, G. Semeraro, P. Lops, and M. de Gemmis, "Exploiting big data for enhanced representations in content-based recommender systems," *Lecture Notes in Business Information Processing*, vol. 152, pp. 182–193, 2013.

[54] Y. Zhang, H. Wu, V. Sorathia, and V. K. Prasanna, "Event recommendation in social networks with linked data enablement.," in *ICEIS (2)* (S. Hammoudi, L. A. Maciaszek, J. Cordeiro, and J. L. G. Dietz, eds.), pp. 371–379, SciTePress, 2013.

[55] N. Kushwaha and O. P. Vyas, "SemMovieRec: Extraction of Semantic Features of DBpedia for Recommender System," in *7th ACM India Computing Conference*, pp. 13:1–13:9, 2014.

[56] T. Di Noia, R. Mirizzi, V. C. Ostuni, and D. Romito, "Exploiting the web of data in model-based recommender systems," in *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, (New York, NY, USA), pp. 253–256, ACM, 2012.

[57] V. C. Ostuni, T. Di Noia, R. Mirizzi, and E. Di Sciascio, *A Linked Data Recommender System Using a Neighborhood-Based Graph Kernel*, pp. 89–100. Cham: Springer International Publishing, 2014.

[58] M. Schmachtenberg, T. Strufe, and H. Paulheim, "Enhancing a location-based recommendation system by enrichment with structured data from the web," in *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, p. 17, ACM, 2014.

[59] G. Rabello Lopes, L. A. Paes Leme, B. Pereira Nunes, and M. A. Casanova, "RecLAK: Analysis and Recommendation of Interlinking Datasets," in *Proceedings of the Workshops at the LAK 2014 Conference* (K. Yacef and H. Drachsler, eds.), (Indianapolis, Indiana, USA), pp. 1–6, CEUR Workshop Proceedings, 2014.

[60] S. Manoj Kumar, K. Anusha, and K. Santhi Sree, "Semantic Web-based Recommendation: Experimental Results and Test Cases," *International Journal of Emerging Research in Management & Technology*, vol. 4, no. 6, pp. 215–222, 2015.

[61] H.-G. Ko, J.-S. Son, and I.-Y. Ko, "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, (New York, NY, USA), pp. 49–50, ACM, 2015.

[62] M. Rowe, "Transferring semantic categories with vertex kernels: Recommendations with semanticSVD++," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8796, pp. 341–356, 2014.

[63] A. Lommatzsch, B. Kille, and S. Albayrak, "Learning hybrid recommender models for heterogeneous semantic data," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13*, SAC '13, (New York, USA), p. 275, ACM Press, 2013.

[64] A. Lommatzsch, B. Kille, and S. Albayrak, "A framework for learning and analyzing hybrid recommenders based on heterogeneous semantic data," in *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, pp. 137–140, LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2013.

[65] D. Torres, H. Skaf-Molli, P. Molli, and A. Díaz, "BlueFinder: recommending wikipedia links using DBpedia properties," in *Proceedings of the 5th Annual ACM Web Science Conference*, WebSci '13, (New York, NY, USA), pp. 413–422, ACM, 2013.

[66] A. Castellanos, A. García-Serrano, and J. Cigarrán, "Linked data-based conceptual modelling for recommendation: a fca-based approach," in *International Conference on Electronic Commerce and Web Technologies*, pp. 71–76, Springer, 2014.

[67] B. Heitmann and C. Hayes, "Using linked data to build open, collaborative recommender systems.," in *AAAI spring symposium: linked data meets artificial intelligence*, pp. 76–81, 2010.

[68] E. Mannens, S. Coppens, T. D. Pessemier, H. Dacquin, D. V. Deursen, R. D. Sutter, and R. V. d. Walle, "Automatic news recommendations via aggregated profiling," *Multimedia Tools and Applications*, vol. 63, pp. 407–425, Mar. 2013.

[69] F. Zarrinkalam and M. Kahani, "A multi-criteria hybrid citation recommendation system based on linked data," in *2012 2nd International eConference on Computer and Knowledge Engineering (ICCKE)*, pp. 283–288, Oct. 2012.

[70] R. Chawuthai, H. Takeda, and T. Hosoya, "Link Prediction in Linked Data of Interspecies Interactions Using Hybrid Recommendation Approach," in *Semantic Technology SE - 9* (T. Supnithi, T. Yamaguchi, J. Z. Pan, V. Wuwongse, and M. Buranarach, eds.), vol. 8943 of *Lecture Notes in Computer Science*, pp. 113–128, Springer International Publishing, 2015.

[71] F. Hopfgartner and J. M. Jose, "Semantic user profiling techniques for personalised multimedia recommendation," *Multimedia Systems*, vol. 16, pp. 255–274, Aug. 2010.

[72] G. Rabello Lopes, L. A. P. Paes Leme, B. Pereira Nunes, M. A. Casanova, and S. Dietze, *Two Approaches to the Dataset Interlinking Recommendation Problem*, pp. 324–339. Cham: Springer International Publishing, 2014.

[73] V. Maccatrozzo, D. Ceolin, L. Aroyo, and P. Groth, "A Semantic Pattern-Based Recommender," in *Semantic Web Evaluation Challenge: SemWebEval 2014 at ESWC 2014*, pp. 182–187, Springer, 2014.

[74] S. Gordea, A. Lindley, and R. Graf, "Computing Recommendations for Long Term Data Accessibility basing on Open Knowledge and Linked Data," in *RecSys 2011 Workshop on Human Decision Making in Recommender Systems affiliated with the 5th ACM Conference on Recommender Systems*, (Chicago, USA), pp. 1–8, 2011.

[75] O. Ozdikis, F. Orhan, and F. Danismaz, "Ontology-based recommendation for points of interest retrieved from multiple data sources," in *Proceedings of the International Workshop on Semantic Web Information Management - SWIM '11*, (New York, New York, USA), pp. 1–6, ACM Press, 2011.

[76] A. Calì, S. Capuzzi, M. M. Dimartino, and R. Frosini, "Recommendation of text tags in social applications using linked data," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8295 LNCS, pp. 187–191, 2013.

[77] A. Corallo, G. Lorenzo, and G. Solazzo, "A semantic recommender engine enabling an etourism scenario," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 1092–1101, Springer, 2006.

[78] R. Mirizzi, T. Di Noia, A. Ragone, V. C. Ostuni, and E. Di Sciascio, "Movie recommendation with dbpedia.," in *Italian Information Retrieval Workshop 2012*, pp. 101–112, Citeseer, 2012.

[79] A. Passant and S. Decker, "Hey! ho! let's go! explanatory music recommendations with dbrec," in *The Semantic Web: Research and Applications* (L. Aroyo, G. Antoniou, E. Hyvönen, A. t. Teije, H. Stuckenschmidt, L. Cabral, and T. Tudorache, eds.), no. 6089 in Lecture Notes in Computer Science, pp. 411–415, Springer Berlin Heidelberg, Jan. 2010.

[80] A. Passant and Y. Raimond, "Combining social music and semantic web for music-related recommender systems," in *CEUR Workshop Proceedings*, vol. 405, 2008.

[81] O. Celma and X. Serra, "FOAFing the music: Bridging the semantic gap in music recommendation," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, pp. 250–256, Nov. 2008.

[82] J. Waitelonis and H. Sack, "Towards exploratory video search using linked data," *Multimedia Tools and Applications*, vol. 59, pp. 645–672, July 2012.

[83] M. Kaminskas, I. Fernández-Tobías, F. Ricci, and I. Cantador, "Knowledge-based music retrieval for places of interest," in *Proceedings of the Second International ACM Workshop on Music Information Retrieval with User-centered and Multimodal Strategies*, MIRUM '12, (New York, NY, USA), pp. 19–24, ACM, 2012.

[84] J. P. Leal, V. Rodrigues, and R. Queirós, "Computing semantic relatedness using dbpedia," in *OASIcs-OpenAccess Series in Informatics*, vol. 21, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

[85] A. Wardhana and H. Nugroho, "Combining FOAF and music ontology for music concerts recommendation on facebook application," in *2013 Conference on New Media Studies (CoNMedia)*, pp. 1–5, Nov. 2013.

[86] A. A. M. Caraballo, N. M. Arruda, B. P. Nunes, G. R. Lopes, and M. A. Casanova, "TRTML - A tripleset recommendation tool based on supervised learning algorithms," in *The Semantic Web: ESWC 2014 Satellite Events SE*, vol. 8798, pp. 413–417, 2014.

[87] A. Lommatzsch, B. Kille, J. W. Kim, and S. Albayrak, "An adaptive hybrid movie recommender based on semantic data," in *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, OAIR '13, (Paris, France, France), pp. 217–218, LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2013.

[88] B. Heitmann, R. Cyganiak, C. Hayes, and S. Decker, "An empirically grounded conceptual architecture for applications on the web of data," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 1, pp. 51–60, 2012.

[89] I. Horrocks, B. Parsia, P. Patel-Schneider, and J. Hendler, "Semantic Web architecture: Stack or two towers?," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3703 LNCS, pp. 37–41, 2005.

[90] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia - A crystallization point for the Web of Data," *Journal of Web Semantics*, vol. 7, no. 3, pp. 154–165, 2009.

[91] M. Schmachtenberg, C. Bizer, and H. Paulheim, "Adoption of the linked data best practices in different topical domains," in *The Semantic Web - ISWC 2014*, vol. 8796 of *Lecture Notes in Computer Science*, pp. 245–260, Springer International Publishing, 2014.

[92] The community project DBpedia, "The DBpedia data set (2014)." `http://wiki.dbpedia.org/Datasets#h434-7`, 2015. Accessed: 2015-01-10.

[93] M. Stankovic, W. Breitfuss, and P. Laublet, "Linked-data based suggestion of relevant topics," in *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, (New York, NY, USA), pp. 49–55, ACM, 2011.

[94] G. Klyne and J. Carroll, "Resource description framework (RDF): Concepts and abstract syntax," W3C recommendation, W3C, Feb. 2004. `http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/`.

[95] M. AlemZadeh, *Semantic Analysis of Wikipedia's Linked Data Graph for Entity Detection and Topic Identification Applications*. PhD thesis, University of Waterloo, 2012.

[96] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, (San Francisco, CA, USA), pp. 448–453, Morgan Kaufmann Publishers Inc., 1995.

[97] N. Seco, T. Veale, and J. Hayes, "An Intrinsic Information Content Metric for Semantic Similarity in WordNet," in *European Conference on Artificial Intelligence*, pp. 1089–1090, 2004.

[98] M. Hadj Taieb, M. Ben Aouicha, M. Tmar, and A. Hamadou, "New information content metric and nominalization relation for a new wordnet-based method to measure the semantic relatedness," in *Cybernetic Intelligent Systems (CIS), 2011 IEEE 10th International Conference on*, pp. 51–58, Sept 2011.

[99] H. Paulheim and J. Fümkranz, "Unsupervised generation of data mining features from linked open data," in *Proceedings of the 2Nd International Conference on Web Intelligence, Mining and Semantics*, WIMS '12, (New York, NY, USA), pp. 31:1–31:12, ACM, 2012.

[100] D. C. Corrales, A. Ledezma, and J. C. Corrales, "A Conceptual Framework for Data Quality in Knowledge Discovery Tasks (FDQ-KDT): A Proposal," *Journal of Computers*, vol. 10, pp. 396–405, nov 2015.

[101] U. Kuzelewska, "Clustering algorithms in hybrid recommender system on MovieLens data," *Studies in Logic, Grammar and Rhetoric*, vol. 37, no. 50, pp. 125–139, 2014.

[102] D. Pelleg and A. Moore, "X-means: Extending K-means with Efficient Estimation of the Number of Clusters," in *17th International Conf. on Machine Learning*, pp. 727–734, 2000.

[103] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 2007.

[104] M. Mishra and H. S. Behera, "Kohonen Self Organizing Map with Modified K-means clustering For High Dimensional Data Set," *International Journal of Applied Information Systems*, vol. 2, no. 3, pp. 34–39, 2012.

[105] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, pp. 411–423, may 2001.

[106] S. Acharyya, O. Koyejo, and J. Ghosh, "Learning to rank with Bregman divergences and monotone retargeting," *Uai*, 2012.

[107] P. Berka, "Data cleansing using clustering," in *Man–Machine Interactions 4*, pp. 391–399, Springer, 2016.

[108] B. Kulis, M. A. Sustik, and I. S. Dhillon, "Low-rank kernel learning with bregman matrix divergences," *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 341–376, 2009.

[109] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with bregman divergences," *Journal of machine learning research*, vol. 6, no. Oct, pp. 1705–1749, 2005.

[110] G. Shani and A. Gunawardana, "Evaluating Recommendation Systems," in *Recommender Systems Handbook SE - 8* (F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, eds.), pp. 257–297, Springer US, 2011.

[111] J. Fleiss *et al.*, "Measuring nominal scale agreement among many raters," *Psychological Bulletin*, vol. 76, no. 5, pp. 378–382, 1971.

[112] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[113] P. Petratos, "Information Retrieval Systems: A Human Centered Approach," *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 2, no. 1, pp. 17–32, 2007.

[114] M. Fleischman and E. Hovy, "Recommendations without user preferences: a natural language processing approach," in *Proceedings of the 8th international conference on Intelligent user interfaces*, pp. 242–244, ACM, 2003.

[115] J. Beel, C. Breitinger, S. Langer, A. Lommatzsch, and B. Gipp, "Towards reproducibility in recommender-systems research," *User Modeling and User-Adapted Interaction*, vol. 26, no. 1, pp. 69–101, 2016.

[116] C. Figueroa, H. Ordoñez, J.-C. Corrales, C. Cobos, L. K. Wives, and E. Herrera-Viedma, "Improving Business Process Retrieval Using Categorization and Multimodal Search," *Knowledge-Based Systems*, 2016.

[117] O. Rodríguez Rocha, I. Vagliano, C. Figueroa, F. Cairo, G. Futia, C. A. Licciardi, M. Marengo, and F. Morando, "Semantic Annotation and Classification In Practice ," *IT Professional*, vol. 17, no. 12 - IT-Enabled Business Innovation, pp. 33–39, 2015.

[118] O. Rodríguez Rocha, C. Figueroa, I. Vagliano, and B. Moltchanov, "Linked Data-Driven Smart Spaces," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems SE - 1* (S. Balandin, S. Andreev, and Y. Koucheryavy, eds.), vol. 8638 of *Lecture Notes in Computer Science*, pp. 3–15, Springer International Publishing, 2014.

[119] T. Heath and C. Bizer, "Linked data: Evolving the web into a global data space," *Synthesis lectures on the semantic web: theory and technology*, vol. 1, no. 1, pp. 1–136, 2011.

[120] World Wide Web Consortium (W3C), "W3c data activity building the web of data." `http://www.w3.org/2013/data/`, 2013. Accessed: 2015-01-10.

[121] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, "DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic Web Journal*, 2014.

[122] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work - CSCW '94*, pp. 175–186, 1994.

[123] Y. Hu, Z. Wang, W. Wu, J. Guo, and M. Zhang, "Recommendation for movies and stars using YAGO and IMDB," in *Web Conference (APWEB), 2010 12th International Asia-Pacific*, pp. 123–129, Apr. 2010.

[124] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," *ACM SIGMOD Record*, vol. 29, pp. 93–104, May 2000.

[125] K. Dhana, "Identify, describe, plot, and remove the outliers from the dataset," 2016.

[126] A. Shahram, "Data Preparation for Predictive Modeling: Resolving Outliers," 2015.

Appendix A

Conceptual Foundations

This appendix deals with the conceptual foundations of the Web of Data and Recommender Systems (RS). It presents a comprehensive overview of the technologies, standards, and principles of the Web of Data, as well as the classification and problems of RS.

A.1 The Web of Data

The *Web of Data* is a subset of the World Wide Web based on the integration of a subset of the Semantic Web technology stack with existing standards of the World Wide Web [15]. Unlike the World Wide Web that consists of human-readable documents linked via hyperlinks, the *"Web of Data"* refers to a global space of structured and machine-readable data[16]. Moreover, while the World Wide Web provides only one type of links known as hyperlinks that are intended to redirect documents, the Web of Data offers different types of links to give a different meaning to each relationship between data, in this way the web is taken to a semantic level where the knowledge about the relationships between data acquires value. For this reason, the web of data is based on open standards for describing, publicizing, interconnecting, and consuming data, even though these data come from different sources [119].

At the present, the web of data is being promoted as the platform for the distribution of data on the Web through the efforts and policies of international institutions such as: the G8 Open Data Charter[1], the executive order of President Obama[2], and the directive PSI of the European Union[3] [120]. For this reason, the web of data

[1] `https://www.gov.uk/government/publications/open-data-charter`

[2] `http://www.whitehouse.gov/blog/2013/05/09/landmark-steps-liberate-open-data`

[3] `http://ec.europa.eu/digital-agenda/en/legal-rules`

has been widespread in various domains such as entertainment (e.g. music, movies), education (e.g. books, scientific bibliography), people, government, among others; making possible to implement general purpose applications operating on different data spaces, and to take advantage of the huge descriptive potential they offer.

A.1.1 Linked Data

In 1994, Tim Berners-Lee[4] uncovered the need of introducing semantics into the Web to extend its capabilities and to publish structured data on it, which became known as *Semantic Web*. The set of good practices or principles for publishing and linking structured data on the Web is known as Linked Data. While the Semantic Web is the goal, Linked Data provides the means to make it reality [3]. The set of Linked Data principles are:

- Use URIs (Uniform Resource Identifier) as names for things.
- Use HTTP URIs, so that people can look up those names.
- Use of standard mechanisms to provide useful information when someone looks up a URI, for example *RDF* (Resource Description Framework) to represent data as graphs and *SPARQL* (SPARQL Protocol and RDF Query Language) to query Linked Data.
- Include links to other URIs, so that they can discover more things.

URIs are a fundamental concept for the Web architecture, intended to increase the value of the World Wide Web trough a "single global identification system" [17]. As constraint URIs should be unique so distinct resources must be assigned distinct URIs.

Resources are objects or concepts identified with a URI. To represent these resources there are various languages, but the most widely used is the RDF language. In this thesis the terms "concept" and "resource" are used indifferently to denote abstract "things" or objects of the real world.

Resource Description Framework

RDF is a recommendation of the W3C that provides a generic graph-based data model for describing resources, including their relationships with other resources [3].

The graph data model of the RDF framework is composed of triples or statements. Each triple contains a subject, a predicate, and an object. Figure A.1 shows

[4] `http://www.w3.org/Talks/WWW94Tim`

an example of the simplest RDF graph containing a subject linked to an object through a predicate.

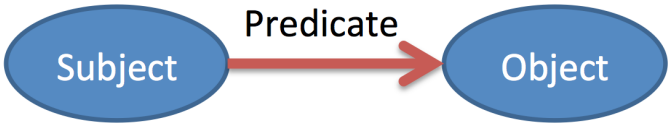


Figure A.1. An RDF graph with two nodes. Figure based on the original shown in the W3C recommendation available on [1]

Triples assert facts about the resources [1]. In a triple the subject is an input resource from which an arc leaves, the predicate is a property (link) that labels the arc, and the object is an output resource or literal (where the arc ends). Literals are resources that can be used for values such as strings, numbers and dates. RDF data can be written in different ways known as serialization e.g. RDF/XML, Notation-3 (N3), Turtle, N-Triples, RDFa, and RDF/JSON [3].

Thanks to the RDF or other languages for describing resources the Linked Data is able use different vocabularies for representing resources such as: people `http://xmlns.com/foaf/spec/`, social media `http://rdfs.org/sioc/spec/`, commerce `http://purl.org/goodrelations/`, events `http://motools.sourceforge.net/event/event.html`, radio and tv programs `http://purl.org/ontology/po/`, among others.

Linked Data datasets

A dataset can be seen as a database storing a collection of triples that may or not belong to a specific domain. More formally, Passant [18] has defined a dataset as: *"A dataset following the Linked Data principles is a graph G such as $G = (R, L, I)$ in which $R = \{r_1, r_2, \ldots, r_n\}$ is a set of resources –identified by their URI–, $L = \{l_1, l_2, \ldots, l_n\}$ is a set of typed links –identified by their URI– and $I = \{i_1, i_2, \ldots, i_n\}$ is a set of instances of these links between resources, such as $i_i = \langle l_j, r_a, r_b \rangle$"*

This definition assumes the interlinked structure of the data in a Linked Data dataset which is not only limited to interlink resources within a dataset, but also made possible to interlink datasets. In this way Linked Data has made possible to create ecosystems of datasets composed of interlinked structured data.

In this sense, one of the most ambitious projects is known as "Linked Open Data" (hereinafter LOD), which corresponds to a joint effort of various international organizations (e.g. the British Broadcasting Corporation, governments of different countries, Thompson Reuters, Flickr among others) to promote the creation of the web of data. This project was initiated in January 2007, supported by the group of education in semantic Web W3C (Web Education and Outreach Group), and aimed

to identify datasets available under open licenses, convert RDF files, and publish web of data [3].

A recent study by Schmachtenberg et al., [91] about the state of Linked Open Data has evidenced the grow of the Linked Open Data from a dozen datasets in 2007 into hundreds of datasets today. Figure A.2 shows a visualization of the overall distribution and graph structure of the Linked Open Data cloud as of January of 2017. A high-resolution and complete visualization of it is available at `http://lod-cloud.net`.

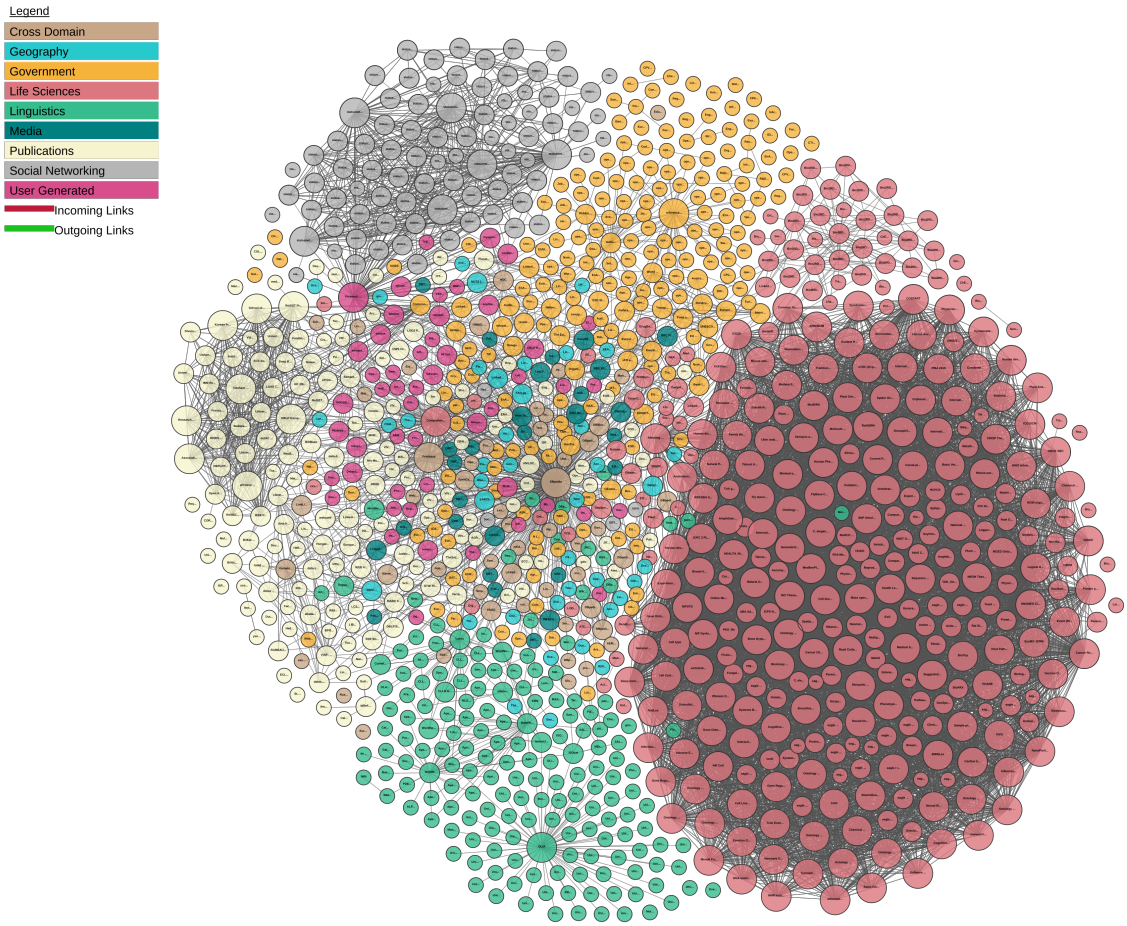


Figure A.2. Linking Open Data cloud diagram 2017, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak. `http://lod-cloud.net/`

Some examples of the most important datasets are: DBpedia, GeoNames, FOAF Profiles, MusicBrainz, WordNet, and DBPLP bibliography.

- DBpedia this is one of the datasets most widely known and used in different applications. It uses a large multi-domain ontology derived from Wikipedia[121]. Currently the english version of DBpedia contains 4.58 million resources and 583 million triples [5].

- GeoNames is a geographical dataset that provides RDF descriptions of geographical resources around the world that contains about 10 million geographical names[6].

- FOAF Profiles is a dataset contains data from personal homepages represented through the FOAF (Friend of a Friend) vocabulary.

- MusicBrainz contains data about artists, tracks, releases and their relationships. The data is extracted from the MusicBrainz music metadata catalogue[7].

- WordNet contains RDF/OWL data that represents the lexical database WordNet[8]

- DBLP is a bibliography dataset that contains data about research papers, authors, conferences, journals among others[9].

It is worth noting that the objective of the web of data is not only to allow applications to discover new datasets following links between them, but also to facilitate the integration of existing data sources to create new applications [119]. In Chapter 2 the use of these datasets in recommender systems for different application domains will be described.

Linked Data endpoints

Endpoints are the mechanism used in Linked Data to provide access to the available datasets. Endpoints may be seen as interfaces to execute queries to the datasets in a similar way as in a database. The language to express query across diverse datasets is SPARQL which is the de facto language for interaction with Linked Data [19]. SPARQL is a language that contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions [20].

---

[5]The updated information about DBpedia can be found on the website of the community project DBpedia at `http://wiki.dbpedia.org/Datasets`

[6]`http://www.geonames.org/about.html`

[7]`https://musicbrainz.org`

[8]`http://wordnet.princeton.edu`

[9]`http://dblp.l3s.de/d2r/`

A SPARQL endpoint accepts queries in SPARQL language and returns results via HTTP in multiple formats that may be: XML, JSON, RDF (RDF, XML, N-Triples, Turtle, etc), and HTML.

The W3C published a comprehensive list[10] of SPARQL and Modena Labs maintains a list of the SPARQL endpoints according to their availability[11].

A.2 Recommender Systems

Recommender Systems (RS) are software tools that use analytic technologies to suggest different items of interest to an end user. These items can belong to different categories or types, e.g. songs, places, news, books, films, events, etc. According to Adomavicius and Tuzhilin [21], the roots of RS can be traced back to the works in cognitive science, approximation theory, information retrieval, forecasting theories, management science, and consumer choice modeling in marketing.

Nowadays, RS are focused on the recommendation problem, which looks for guiding users in a personalized way to interesting items in a large space of possible options [7]. Typically RS are classified as: content-based, collaborative filtering, knowledge-based, and hybrid [2].

A.2.1 Classification of Recommender Systems

Content-based (CB) RS make suggestions taking into account the ratings that users give to items according to their preferences and considering also the content of these items (e.g. keywords, title, pixels, disk space, etc) [7]. In other words, these RS suggest similar items with those the user preferred in the past [21]. Analyzed content may be directly derived from the item itself, for example the keywords extracted from a text, pixels of an image, track tempo, etc, or derived from item metadata, for instance the publication year of a book, its author, the number of pages etc [22]. The syntactic nature of the CB based on the similarities between items sharing the same set of attributes of features provides recommendations overspecialized that include only items that are really similar to those that the user already knew as similar [12].

Collaborative-filtering (CF) RS are the recommenders most mature and widely adopted due to their good results and their easy implementation [22]. According to Park et al., [12] the origin of RS goes back to the work of Resnick et al.,[122] which the technique of recommendation, named “collaborative filtering” is described. This

[10] `http://www.w3.org/wiki/SparqlEndpoints`

[11] `http://labs.mondeca.com/sparqlEndpointsStatus/index.html`

technique generates recommendations of items to a user taking into account ratings that users with similar preferences have given to these items [8].

Knowledge-based RS infer and analyze similarities between user requirements and features of items described in a knowledge base that models users and items according to a specific application domain [9]. Afterwards, the knowledge base is used to apply inference techniques to find similarities between user needs and items features. This type of requires an information base including knowledge about the items to be recommended as well as the needs and preferences of users [22].

Hybrid RS combine one or more of the aforementioned techniques in order to circumvent limitations of individual techniques. According to Felfernig et al., [8] these combinations may be performed in the following ways: implementing two types of RS separately and combine their results; adding features form KB recommenders to CF; and developing a unique RS integrating both techniques relying on probabilistic and statistical tools.

A.2.2 Recommender Systems and Linked Data

With the evolution of the Web towards a global space of connected and structured data, a new kind of knowledge-based RS has emerged known as Linked Data-based RS. This kind of RS suggest items taking into account the knowledge of datasets published under the Linked Data principles. The main benefit of using Linked Data as a source for generating recommendations is the large amount of available concepts and their links that can be used to infer relationships more effectively in comparison to derive the same kind of relationships from text [10].

As Linked Data information is machine readable it is possible to query datasets on a fine-grained level in order to collect information without having to take manual actions, therefore information is explicitly represented. This allows recommender systems to apply reasoning techniques when querying datasets and make implicit knowledge explicit. A complete state of the art of Linked data based RS is presented in chapter 2, which classify the current approaches for recommendation based on Linked Data as well as the algorithms they use.

Approaches for RS based on Linked Data

Chapter 2 presents a classification of the RS according to the way they used Linked Data to produce recommendations and grouped them into: *Linked Data driven RS* that rely solely on the knowledge of the Linked Data to provide recommendations. *Hybrid RS* that exploit Linked Data to perform some operations that can be used or not used to provide recommendations. *Representation only*, which are RS that exploit the RDF format to represent data and use at least one vocabulary or ontology to express the underlying semantics, but Linked Data are not used to provide recommendations. *Exploratory search*, which are not RS, but their main duty is

to assist users to explore knowledge and to suggest relevant to a topic or concept. Probabilistic algorithms

Algorithms for RS based on Linked Data

Additionally, chapter 2 presents a classification of the algorithms implemented in RS based on Linked Data. Some of these classes described in chapter 2 are: *graph-based algorithms* that take advantage of the graph structure of the Linked Data where nodes are concepts are and edges their relationships. *Machine Learning algorithms*, supervised and unsupervised algorithms used in conjunction with Linked Data to generate recommendations. *Memory based algorithms* that operate over specialized matrixes (user-user, user-item, item-attribute, item-item) for similarity calculations. *Probabilistic algorithms* which use probabilities computed based on the notion of estimating a probability of relevance between a pair query and candidate concept and ranking resources in descent order of probability of relevance. *Semantic reasoning algorithms* that use ontologies and reasoning engines in order to calculate semantic distances between concept. *Evolutionary algorithms* inspired on biological evolution in order to generate solutions (recommendations) for an optimization problem.

Appendix B

Selected and Excluded Papers

B.1 Selected Papers for the SLR

This section presents the set of papers selected to conduct the Systematic Literature Review reported in Chaper 2. Table B.1 shows these papers, rows in italics identify papers (P) belonging to a study (S) already reported by other paper (e.g. papers 10, 19, 54 belong to the same study S10).

P	S	Authors	Year	Title	Publication details
1	S1	Fernández-Tobías, I., Cantador, I., Kaminskas, M., Ricci, F.	2011	A generic semantic-based framework for cross-domain recommendation	2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems - HetRec '11, pp 25 - 32
2	S2	Kabutoya, Y., Sumi, R., Iwata, T., Uchiyama, T., Uchiyama, T.	2012	A Topic Model for Recommending Movies via Linked Open Data	International Conferences on Web Intelligence and Intelligent Agent Technology, pp 625 - 630
3	S3	Dell'Aglio, D., Celino, I., Cerizza, D.	2010	Anatomy of a Semantic Web-enabled Knowledge-based Recommender System	4th international workshop Semantic Matchmaking and Resource Retrieval in the Semantic Web, at the 9th International Semantic Web Conference, pp 115 - 130
4	S4	Mannens, E., Coppens, S., Wica, I., Dacquin, H., Van De Walle, R.	2013	Automatic News Recommendations via aggregated Profiling	Journal Multimedia Tools and Applications, 63 (2), pp 407 - 425
5	S5	Dzikowski, J., Kaczmarek, M.	2012	Challenges in Using Linked Data within a Social Web Recommendation Application to Semantically Annotate and Discover Venues	International Cross Domain Conference and Workshop, pp 360 - 374

6	S6	Wardhana, A.T.A.; Nugroho, H.T.	2013	Combining FOAF and Music Ontology for Music Concerts Recommendation on Facebook Application	Conference on New Media Studies, pp 1 - 5
7	S7	Passant, A., Raimond, Y.	2008	Combining Social Music and Semantic Web for music-related recommender systems	First Workshop on Social Data on the Web, pp 19 -30
8	S8	Lindley, A., Graf, R.	2011	Computing Recommendations for Long Term Data Accessibility basing on Open Knowledge and Linked Data	5th ACM Conference on Recommender Systems, pp 51 - 58
9	S9	Passant, Alexandre	2010	dbrec - Music Recommendations Using DBpedia	The Semantic Web - ISWC 2010, pp 209 - 224
10	S10	Stankovic, M., Breitfuss, W., Laublet, P.	2011	Discovering Relevant Topics Using DBPedia: Providing Non-obvious Recommendations	2011 International Conferences on Web Intelligence and Intelligent Agent Technology, 1, pp 219 - 222
11	S11	Marie, N., Gandon, F., Ribière, M., Rodio, F.	2013	Discovery Hub : on-the-fly linked data exploratory search	9th International Conference on Semantic Systems, pp 17 - 24
12	S12	Peska, L., Vojtas, P.	2013	Enhancing Recommender System with Linked Open Data	10th International Conference on Flexible Query Answering Systems, pp 483 - 494
13	S13	Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D.	2012	Exploiting the web of data in model-based recommender systems	6th ACM conference on Recommender systems
14	S14	Golbeck, J.	2006	Filmtrust: movie recommendations from semantic web-based social networks	3rd IEEE Consumer Communications and Networking Conference, pp 1314 - 1315
15	S15	Celma, Ò., Serra, X.	2008	FOAFing the music: Bridging the semantic gap in music recommendation	Web Semantics: Science, Services and Agents on the World Wide Web, 6 (4), 250 - 256
16	S16	Varga, B., Groza, A.	2011	Integrating DBpedia and SentiWordNet for a tourism recommender system	7th International Conference on Intelligent Computer Communication and Processing, pp 133 - 136
17	S17	Kaminskas, M., Fernández-Tobías, I., Ricci, F., Cantador, I.	2012	Knowledge-based music retrieval for places of interest	Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies - MIRUM '12, pp 19 - 24
18	S18	Dietze, S.	2012	Linked Data as facilitator for TEL recommender systems in research & practice	2nd Workshop on Recommender Systems for Technology Enhanced Learning, pp 7 - 10

19	*S10*	*Damljanovic, D., Stankovic, M., Laublet, P.*	*2012*	*Linked Data-Based Concept Recommendation : Comparison of Different Methods*	*9th Extended Semantic Web Conference, pp 24 - 38*
20	S19	Kitaya, K., Huang, H. H., Kawagoe, K.	2012	Music curator recommendations using linked data	Second International Conference on the Innovative Computing Technology, pp 337 - 339
21	S20	Jung, K., Hwang, M., Kong, H., Kim, P.	2005	RDF Triple Processing Methodology for the Recommendation System Using Personal Information	International Conference on Next Generation Web Services Practices, pp 241 - 246
22	S21	Calì, A., Capuzzi, S., Dimartino, M. M., Frosini, R.	2013	Recommendation of Text Tags in Social Applications Using Linked Data	ICWE 2013 Workshops
23	*S21*	*Calì, A., Capuzzi, S., Dimartino, M. M., Frosini, R.*	*2013*	*Recommendation of Text Tags Using Linked Data*	*3rd International Workshop on Semantic Search Over the Web, pp 1 - 3*
24	S22	Meymandpour, R., Davis, J. G.	2012	Recommendations using linked data	5th Ph.D. workshop on Information and knowledge - PIKM '12, pp 75 - 82
25	S23	Harispe, S., Ranwez, S., Janaqi, S., Montmain, J.	2013	Semantic Measures Based on RDF Projections: Application to Content-Based Recommendation Systems	On the Move to Meaningful Internet Systems: OTM 2013 Conferences SE - 44, pp 606 - 615
26	S24	Hopfgartner, F., Jose, J. M.	2010	Semantic user profiling techniques for personalised multimedia recommendation	Multimedia Systems, 16 (4-5), pp 255 - 274
27	*S5*	*Łazaruk, S., Dzikowski, J., Kaczmarek, M., Abramowicz, W.*	*2012*	*Semantic Web Recommendation Application*	Federated Conference on Computer Science and Information Systems (FedCSIS), pp 1055 - 1062
28	S25	Ostuni, V. C., Di Noia, T., Di Sciascio, E., Mirizzi, R.	2013	Top-N recommendations from implicit feedback leveraging linked open data	Proceedings of the 7th ACM conference on Recommender systems, pp 85 - 92
29	S26	Ahn, J., Amatriain, X.	2010	Towards Fully Distributed and Privacy-Preserving Recommendations via Expert Collaborative Filtering and RESTful Linked Data	International Conference on Web Intelligence and Intelligent Agent Technology, pp 66 - 73
30	S27	Heitmann, B., Hayes, C.	2010	Using Linked Data to Build Open , Collaborative Recommender Systems	AAAI Spring Symposium: Linked Data Meets Artificial Intelligence, pp 76 - 81

31	S28	Zarrinkalam, F., Kahani, M.	2012	A multi-criteria hybrid citation recommendation system based on linked data	2nd International eConference on Computer and Knowledge Engineering (ICCKE), 2012, pp 283 - 288
32	S29	Lommatzsch, A., Kille, B., Kim, J. W., Albayrak, S.	2013	An Adaptive Hybrid Movie Recommender based on Semantic Data	10th Conference on Open Research Areas in Information Retrieval, pp 217 - 218
33	S30	Torres, D., Skaf-Molli, H., Molli, P.; Díaz, A.	2013	BlueFinder: Recommending Wikipedia Links Using DBpedia Properties	5th Annual ACM Web Science Conference, pp 413 - 422
34	S31	Ostuni, V. C., Di Noia, T., Mirizzi, R., Romito, D., Di Sciascio, E.	2012	Cinemappy : a Context-aware Mobile App for Movie Recommendations boosted by DBpedia	International Workshop on Semantic Technologies meet Recommender Systems & Big Data SeRSy 2012, pp 37 - 48
35	S33	Zhang, Y. Wu, H. Sorathia, V., Prasanna, V. K.	2008	Event recommendation in social networks with linked data enablement	15th International Conference on Enterprise Information Systems, pp 371 - 379
36	S34	Mirizzi, R., Di Noia, T.	2010	From exploratory search to web search and back	3rd workshop on Ph.D. students in information and knowledge management - PIKM '10, pp 39 - 46
37	S35	Khrouf, H., Troncy, R.	2013	Hybrid event recommendation using linked data and user diversity	Proceedings of the 7th ACM conference on Recommender systems, pp 185 - 192
38	S36	Bahls, D., Scherp, G., Tochtermann, K., Hasselbring, W.	2012	Towards a Recommender System for Statistical Research Data	2nd International Workshop on Semantic Digital Archives
39	S37	Cheng, Gong; Gong, Saisai; Qu, Yuzhong	2011	An Empirical Study of Vocabulary Relatedness and Its Application to Recommender Systems	10th International Conference on The Semantic Web - Volume Part I, pp 98 - 113
40	S38	Wang, Y., Stash, N., Aroyo, L., Gorgels, P., Rutledge, L., Schreiber, G.	2008	Recommendations based on semantically enriched museum collections	Web Semantics: Science, Services and Agents on the World Wide Web, 6 (4), 283 - 290
41	*S11*	*Marie, N., Gandon, F., Legrand, D., Ribière, M.*	*2015*	*Discovery Hub: a discovery engine on the top of DBpedia*	*3rd International Conference on Web Intelligence, Mining and Semantics*
42	S31	Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D., Zanker, M.	2012	Linked open data to support content-based recommender systems	8th International Conference on Semantic Systems
43	*S31*	*Ostuni, Vito Claudio; Gentile, Giosia; Noia, Tommaso Di; Mirizzi, Roberto; Romito, Davide; Sciascio, Eugenio Di*	*2015*	*Mobile Movie Recommendations with Linked Data*	*International Cross-Domain Conference, pp 400 - 415*

44	*S31*	*Mirizzi, R., Di Noia, T., Ragone, A., Ostuni, V. C., Di Sciascio, E.*	*2012*	*Movie recommendation with DBpedia*	*3rd Italian Information Retrieval Workshop, pp 101 - 112*
45	S39	Waitelonis, J., Sack, H.	2011	Towards exploratory video search using linked data	Multimedia Tools and Applications, 59 (2), pp 645 - 672
46	S40	Li, S., Zhang, Y., Sun, H.	2010	Mashup FOAF for Video Recommendation LightWeight Prototype	7th Web Information Systems and Applications Conference, pp 190 - 193
47	S41	Hu, Y., Wang, Z., Wu, W., Guo, J., Zhang, M.	2010	Recommendation for Movies and Stars Using YAGO and IMDB	12th International Asia-Pacific Web Conference, pp 123 - 129
48	S42	Ruotsalo, T., Haav, K., Stoyanov, A., Roche, S., Fani, E., Deliai, R., Mäkelä, E., Kauppinen, T., Hyvönen, E.	2013	SMARTMUSEUM: A mobile recommender system for the Web of Data	Web Semantics: Science, Services and Agents on the World Wide Web, 20, pp 50 - 67
49	S43	Stankovic, M., Jovanovic, J., Laublet, P.	2011	Linked Data Metrics for Flexible Expert Search on the Open Web	8th Extended Semantic Web Conference, pp 108 - 123
50	S44	Ozdikis, O., Orhan, F., Danismaz, F.	2011	Ontology-based recommendation for points of interest retrieved from multiple data sources	International Workshop on Semantic Web Information Management, pp 1 - 6
51	S45	Debattista, J., Scerri, S., Rivera, I., Handschuh, S.	2012	Ontology-based rules for recommender systems	International Workshop on Semantic Technologies meet Recommender Systems & Big Data, pp 49 - 60
52	S46	Codina, V.; Ceccaroni, L.	2010	Taking Advantage of Semantics in Recommendation Systems	2010 Conference on Artificial Intelligence Research and Development, pp 163 - 172
53	*S9*	*Passant, A., Decker, S.*	*2010*	*Hey! Ho! Let's Go! Explanatory Music Recommendations with dbrec*	*7th Extended Semantic Web Conference, pp 411 - 415*
54	*S10*	*Stankovic, M., Breitfuss, W., Laublet, P.*	*2011*	*Linked-data based suggestion of relevant topics*	*7th International Conference on Semantic Systems, pp 49 - 55*
55	*S9*	*Passant, A.*	*2010*	*Measuring semantic distance on linking data and using it for resources recommendations*	*AAAI Spring Symposium: Linked Data Meets Artificial Intelligence, pp 93 - 98*
56	*S14*	*Golbeck, J.*	*2006*	*Generating Predictive Movie Recommendations from Trust in Social Network*	*4th International Conference, iTrust 2006, pp 93 - 104*
57	*S39*	*Sack, H.*	*2009*	*Augmenting Video Search with Linked Open Data*	*International Conference on Semantic Systems, pp 550 - 558*

58	S47	Baumann, S., Schirru, R., Streit, B.	2011	Towards a Storytelling Approach for Novel Artist Recommendations	8th International Workshop, AMR 2010, Linz, Austria, August 17-18, 2010, Revised Selected Papers, pp 1 - 15
59	S48	Corallo, A., Lorenzo, G., Solazzo, G.	2006	A Semantic Recommender Engine Enabling an eTourism Scenario	10th International Conference, pp 1092 - 1101
60	S49	Nuzzolese, A. G., Presutti, V., Gangemi, A., Musetti, A., Ciancarini, P.	2013	Aemoo: Exploring Knowledge on the Web	Proceedings of the 5th Annual ACM Web Science Conference, pp 272 - 275
61	*S49*	*Musetti, A., Nuzzolese, A., Draicchio, F., Presutti, V., Blomqvist, E., Gangemi, A., Ciancarini, P.*	*2012*	*Aemoo: Exploratory Search based on Knowledge Patterns over the Semantic Web*	*Semantic Web Challenge*
62	*S47*	*Baumann, S., Schirru, R.*	*2012*	*Using Linked Open Data for Novel Artist Recommendations*	*13th Internal Society for Music Information Retrival Conference*
63	S50	Cantador, I., Castells, P.	2006	Multilayered Semantic Social Network Modeling by Ontology-Based User Profiles Clustering: Application to Collaborative Filtering	Proceedings of 15th International Conference, pp 334 - 349
64	*S34*	*Mirizzi, R., Ragone, A., Di Noia, T., Di Sciascio, E.*	*2010*	*Ranking the Linked Data: The Case of DBpedia*	*10th International Conference, pp 337 - 354*
65	S51	Heitmann, B.,Hayes, C.	2010	Enabling Case-Based Reasoning on the Web of Data	The WebCBR Workshop on Reasoning from Experiences on the Web at International Conference on Case-Based Reasoning
66	S52	Alvaro, G., Ruiz, C., Córdoba, C., Carbone, F., Castagnone, M., Gómez-Pérez, J. M., Contreras, J.,	2011	miKrow : Semantic Intra-enterprise Micro-Knowledge Management System	8th Extended Semantic Web Conference, pp 154 - 168
67	*S50*	*Cantador, I., Castells, P., Bellogín, A.*	*2011*	*An Enhanced Semantic Layer for Hybrid Recommender Systems: Application to News Recommendation*	*Int. J. Semant. Web Inf. Syst., 7 (1), pp 44 - 78*
68	S32	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), pp 1 - 15

69	S29	*Lommatzsch, A., Kille, B., Albayrak, S.*	*2015*	*A Framework for Learning and Analyzing Hybrid Recommenders based on Heterogeneous Semantic Data Categories and Subject Descriptors*	*10th Conference on Open Research Areas in Information Retrieval, pp 137 - 140*

Table B.1: Selected papers (P) for the Systematic Review and corresponding studies (S)

B.2 Papers Excluded from the Systematic Review During the Data Extraction

This section presents the set of papers excluded from the Systematic Literature Review reported in Chapter 2 (Table B.2).

P	Authors	Year	Title	Publication details	Reason
1	Wu, C., Wu, J., Ye, G., He, L., Huang, L., Xie, M.	2013	Linked Course Data-based User Personal Knowledge Recommendation Engine Architecture of Linked Course Data	Journal of Computational Information Systems, 9 (5), pp 1735 - 1742	The study is not a RS
2	Pereira Nunes, B. P., Dietze, S., Casanova, M., A., Kawase, R., Fetahu, B., Nejdl, W.	2013	Combining a Co-occurrence-Based and a Semantic Measure for Entity Linking	10th International Conference, pp 548 - 562	The study is not a RS.
3	Ruotsalo, T., Hyvönen, E.	2007	A Method for Determining Ontology-Based Semantic Relevance	18th International Conference, pp 680 - 688	The study is not a RS. It is a semantic similarity method.
4	Parundekar, R., Oguchi, K.	2012	Driver recommendations of POIs using a semantic content-based approach	International Workshop on Semantic Technologies Meet Recommender Systems and Big Data	The study is a RS but use classical kNN algorithm for recommendation. RDF data used only in an intermediate step to integrate data.

5	Song, T., Zhang, D., Shi, X., He, J.,Kang, Q.	2014	Combining Fusion and Ranking on Linked Data for Multimedia Recommendation	Proceedings of the 9th International Symposium on Linear Drives for Industry Applications, Volume 3, pp 531-538	The full paper of the study not available
6	Pham, X.H., Jung, J.J., Takeda, H.	2013	Exploiting linked open data for attribute selection on recommendation systems	Find out how to access preview-only content	The full paper of the study not available
7	Kim, T., Kim, P., Lee, S., Jung, H., Sung, W. K.	2013	OntoURIResolver: Resolution and recommendation of URIs Published in LOD	Proceedings of the 9th International Symposium on Linear Drives for Industry Applications, Volume 3	The study is not a RS.
8	Kurz, T., BÃ¼rger, T., Sint, R., Mika, P., Vallet, D., Carrero, F. M.	2010	R3-A related resource recommender	Lecture Notes in Electrical Engineering Volume 272, 2014, pp 531-538	The study is not a RS. It is more an interlinking framework
9	Morshed, A., Dutta, R., Aryal, J.	2013	Recommending environmental knowledge as linked open data cloud using semantic machine learning	29th International Conference on Data Engineering Workshops	The study is a knowledge base based on data integration and knowledge recommendation. Linked Data is not used to provide recommendations, but only for data integration.
10	Wu, J.Y., Wu, C.L.	2014	The Study of User Model of Personalized Recommendation System Based on Linked Course Data	Applied Mechanics and Materials, 519-520, pp 1609-1612	The full paper of the study not available (excluded before data extraction)
11	Kim, T., Kim, P., Lee, S., Jung, H., Sung, W. K.	2011	OntoURIResolver: URI Resolution and Recommendation Service Using LOD	Future Generation Information Technology Conference, pp 245 - 250	The study is not a RS.
12	Bianchini, Devis	2012	A Classification of Web API Selection Solutions over the Linked Web	2nd International Workshop on Semantic Search over the Web	The study is not a RS.

13	Bellekens, P., Houben, G.-J., Aroyo, L., Schaap, K., Kaptein, A.	2009	User Model Elicitation and Enrichment for Context-sensitive Personalization in a Multiplatform Tv Environment	Proceedings of the seventh european conference on European interactive television conference, pp 119 - 128	The study is not a RS.
14	Zarrinkalam, F., Kahani, M.	2011	Improving bibliographic search through dataset enrichment using Linked Data	1st International eConference on Computer and Knowledge Engineering, pp 254 - 259	The study is not a RS. It uses Linked Data to enrich data.
15	Sheng, H., Chen, H., Yu, T., Feng, Y.	2010	Linked data based semantic similarity and data mining	2010 IEEE International Conference on Information Reuse & Integration, pp 104 - 108	The study is not a RS. It is a semantic similarity method.
16	Qing, H., Dietze, S., Giordano, D., Taibi, D., Kaldoudi, E., Dovrolis, N.	2012	Linked education : interlinking educational resources and the web of data	27th Annual ACM Symposium on Applied Computing, pp 366 - 371	The study is not a RS. It uses Linked Data to enrich data.
17	Albertoni, R., De Martino, M.	2006	Semantic Similarity of Ontology Instances Tailored on the Application Context	OTM Confederated International Conferences, CoopIS, DOA, GADA, and ODBASE 2006. Proceedings, Part I, pp 1020 - 1038	The study is not a RS. It is a semantic similarity method.
18	Tous, R., Delgado, J.	2006	A Vector Space Model for Semantic Similarity Calculation and OWL Ontology Alignment	17th International Conference, pp 307 - 316	The study is not a RS. It is a semantic similarity method.
19	Leal, J. P., Rodrigues, V., Queirós, R.	2012	Computing Semantic Relatedness using DBPedia	1st Symposium on Languages, Applications and Technologies, pp 133 - 147	The study is not a RS. It is a semantic similarity method.
20	Golbeck, J., Hendler, J.	2006	FilmTrust: movie recommendations using trust in web-based social networks	Consumer Communications and Networking Conference, pp 282 - 286	The study is a RS but not use Linked Data to provide recommendations
21	Xie, M.	2011	Semantic-Based Linked Data Mining and Services	Journal of Information and Computational Science, 12 (December), pp 3981 -3988	The study is not a RS.

22	Shabir, N., Clarke, C.	2009	Using Linked Data as a basis for a Learning Resource Recommendation System	1st International Workshop on Semantic Web Applications for Learning and Teaching Support in Higher Education	The study is not a RS.

Table B.2: Papers Excluded from the Systematic Review During the Data Extraction

Appendix C

User Interfaces for the *AlLied* framework

This appendix shows examples of the graphic user interfaces presented as results for the AlLied framework.

C.1 Mobile application that accesses to the RESTFul interface

Figure C.1 shows the graphical interface of a mobile application that accesses to the RESTful interface. In this application a "mind map" is presented based on the information provided from the recommender.

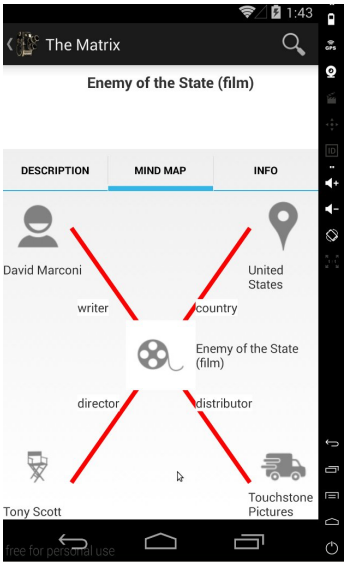


Figure C.1. Example of an application using the RESTful interface to provide film recommendations on a mobile application

C.2 Desktop Application

Figure C.2 presents the main user interfaces for the desktop application. This interfaces allows the user to choose the generator algorithms (Figure C.3) as well as the ranking algorithms (Figure C.4).

Figure C.2. Main GUI for the desktop application

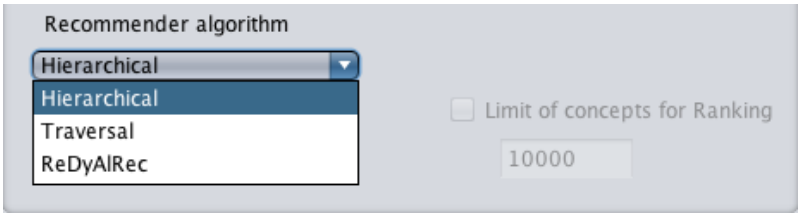


Figure C.3. GUI for choosing generation algorithms

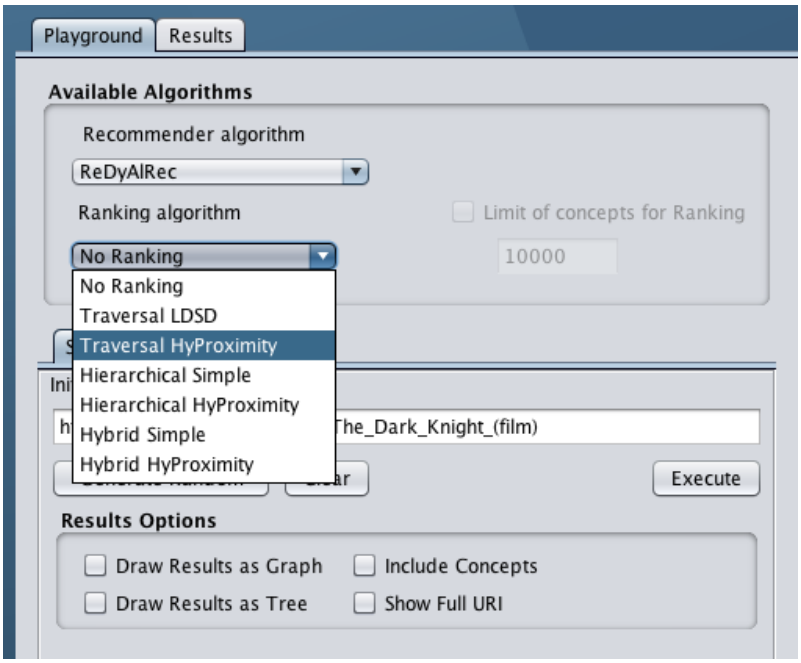


Figure C.4. GUI for choosing ranking algorithms

C.2.1 Tree structure view

Figure C.5 shows an example of the candidate resources for the initial resource *<http://dbpedia.org/resource/Mole_Antonelliana>* in a folder system structure. In this view folder nodes are categories while leaf nodes are candidate resources recommended. This kind of interface facilitates the user to choose the categories of interest in order to obtain a set of results arranged according to the context or domain of relevance.

C.2.2 Graph view

Figure C.6 shows the results for *<http://dbpedia.org/resource/Mole_Antonelliana>* in a graph structure. White nodes are candidate concepts, blue nodes are categories, and the remaining yellow node is the initial concept.

C.3 Web Client Application

This section shows an example of the Web Client Application which is available at `http://natasha.polito.it/AlliedWI/`. Figure C.7 shows an example of the Web Application where users can choose the recommendation algorithm. Figure C.8 presents an example of the results for the web application. The web application only contains some of the algorithms implemented because it is only an example of how to implement a user interface for the AlLied framework. Currently, a more complete web interface is being developed to be similar as the desktop version.

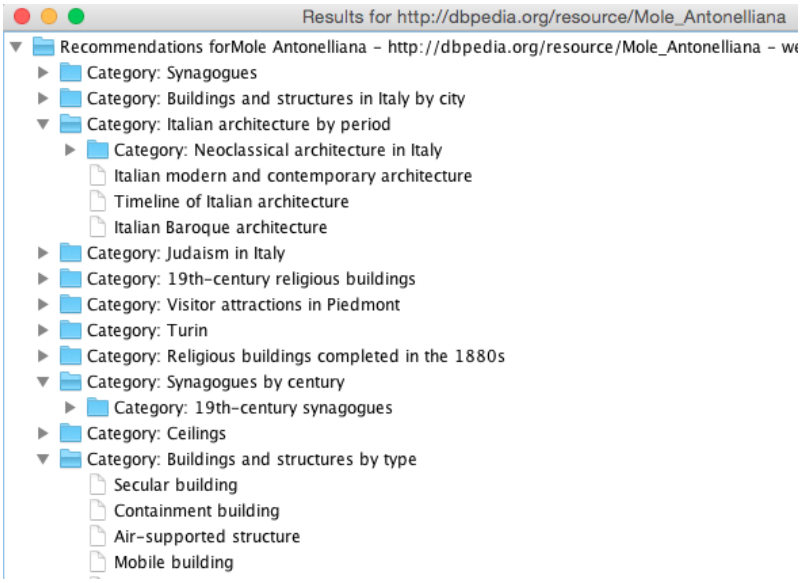


Figure C.5. Example of Recommendations as folder system

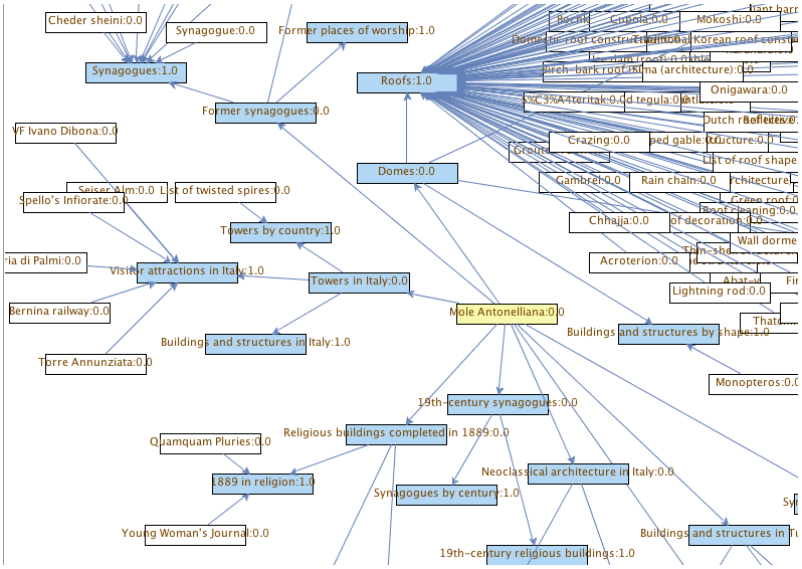


Figure C.6. Example of recommendations as graph

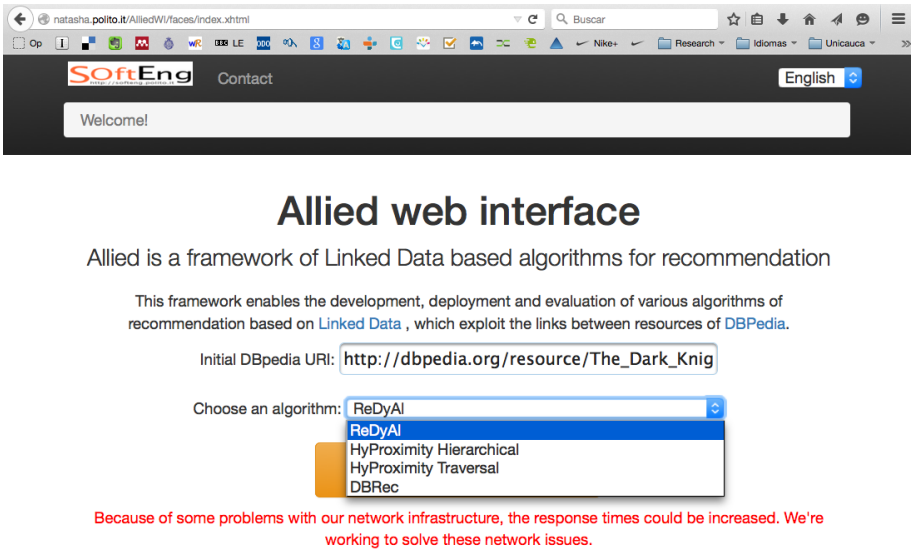
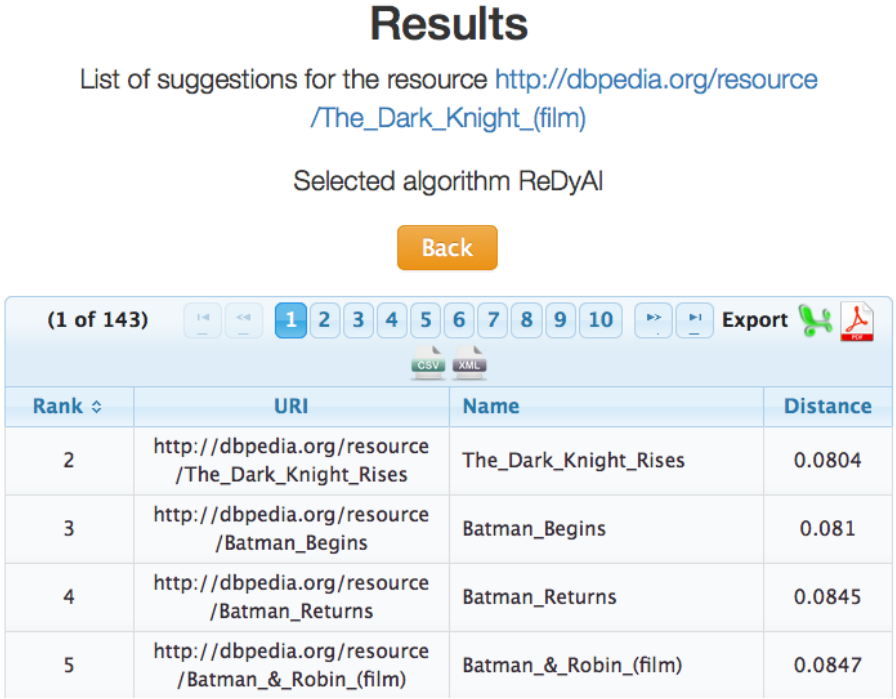


Figure C.7. Home page of a Web Application for the AlLied framework



Rank	URI	Name	Distance
2	http://dbpedia.org/resource/The_Dark_Knight_Rises	The_Dark_Knight_Rises	0.0804
3	http://dbpedia.org/resource/Batman_Begins	Batman_Begins	0.081
4	http://dbpedia.org/resource/Batman_Returns	Batman_Returns	0.0845
5	http://dbpedia.org/resource/Batman_&_Robin_(film)	Batman_&_Robin_(film)	0.0847

Figure C.8. Example of results of the Web Application

Appendix D

Complementary Documentation for the Machine-Learning based Implementation

D.1 Common features for films in RS

The selection of common features for films was conducted by choosing both RS applications and research papers. Table D.1 shows the features for the RS selected from the state of the art.

Features for films described in both RS and papers were extracted and then the most frequently and significant for this study were selected. Features with different name but referring to the same concept were merged in one concept, e.g., subject, category and genre were merged into the feature "subject"; starring and actor in "actor"; etc.

The final set of features selected for the LODMatrix was: name, countries, directors, producers, starring, writers, awards, year, and subject.

- *name:* the title of the film
- *countries:* countries where the film was released.
- *directors:* directors of the film.
- *producers:* producers of the film.
- *starring:* actors of the film.
- *writers:* original writers of the film.
- *year:* year of release.
- *subject:* set of categories of the film.

D.2 LODMatrixes

LODMatrix is a set of matrixes created in this thesis to test the feasibility of Machine Learning algorithms to execute tasks for RS. Table D.2 shows the types of LODMatrix created in this

Name	Type	Features	Reference
Discovery Hub	Explorative Search System	Country, Music Composer, Starring, Writer, Producer, Distributor, Editing, Cinematography, Narrator	http://discoveryhub.co and [40]
LinkedMDB	LD dataset	Actor, date, director, featured_film_location, distributor, music_contributor, performance, producer, runtime, title, writer	http://linkedmdb.org
SemMovieRec: Extraction of semantic features of dbpedia for recommender system	Extractor of features for RS	Writer, director, genres, producer, musicComposer, starring, subject (categories),	[55]
Cinemappy	RS	Starring, director, subject,	[57]
Recommendations and object discovery in graph databases using path semantic analysis	RS	Subject, broader, director, type, producer, country, writer,	[48]
Recommendation for Movies and Stars using YAGO and IMDB	RS	genre, director, writer and cast.	[123]
Learning hybrid recommender models for heterogeneous semantic data	Comparative study of RS	Genere, actor, location, director, country	[63]

Table D.1. Features for RS selected from the state of the art

research for testing purposes. These matrixes may be used for further research with machine learning algorithms for Linked Data based recommendations[1].

Additionally, a database containing the LODMatrix data was created to easy the searching for films, and to generate the various types of LODMatrixes. The entity-relationship diagram is presented in figure D.1.

[1]Copies of the LODMatrixes have been uploaded at https://goo.gl/dCxiDh

ID	Datasets	Description
1	LODMatrixFull	It is a dataset with all possible combinations of the attributes of each film. Attributes: name, countries, directors, producers, starring, writers, years, and subjects
2	LODMatrixCount	It is a dataset where each film is a row, and each cell is the frequency of each attribute (column title) in the film. Attributes: idfilm, countries, directors, producers, actors, writers, and subjects
3	LODMatrixAttributeNameBinary	A LODMatrix(*AtributeName*)Binary is a type of matrixes where each attribute (*AttributeName*) of the film is a column and each cell contains 1 if the attribute is referenced by the film and 0 otherwise. The datasets here are: lodmatrix_actor_binary, LODMatrix_country_Binary, LODMatrix_director_Binary, LODMatrix_producer_Binary, and lodmatrix_subject_binary
4	lodmatrix_allproperties_binary	This dataset gathers all the attributes (actor, country, director, producer, subject, year and name) into a single matrix whose columns are properties, and cells are 1 or 0 depending on whether the movie has the property designated by the column. Only the most frequently properties were selected
5	LODMatrixCatBinary	In this dataset columns are the categories of the films and cells are 1 if the film is in that category and 0 if not - In this case only those categories with frequency greater than 500 were considered. (Frequency of ocurrence in the whole dataset)
6	LODMatrixTestRecRedyAl	It is a dataset containing the recommendation lists generated by ReDyAl along with the voting of the human evaluators. Attributes: idqueryfilm, idrecfilm, relevance (rating that users assigned to ReDyAl recommendations)
7	LODMatrixTestRecommendation	This dataset contains the recommended films (idfilmrec) with corresponding query films (idfilmquery). Attributes: idqueryfilm, idrecfilm, relevance
8	LODMatrixFilmProperty	Contains the total list of movies along with the properties they contain and the type of property Fields: idfilm, idprop, link (link is the property type, it can be: actor, country, director, producer, subject, writer)

Table D.2. Types of LODMatrix datasets created in this thesis

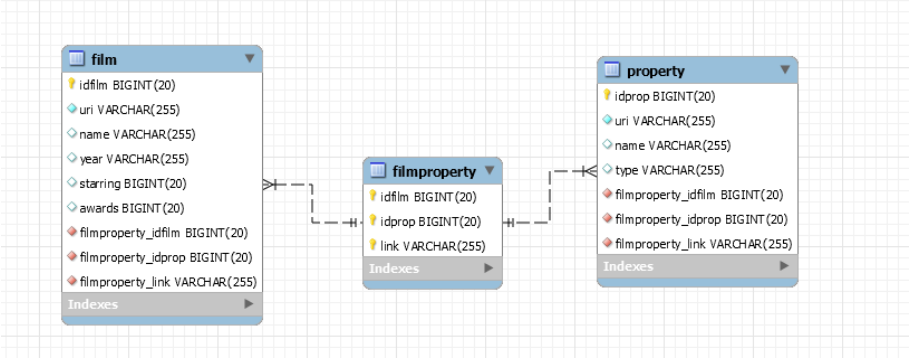


Figure D.1. Entity relationship diagram for the LODMatrixDB

D.3 Outliers Detection

Outliers are observations which deviate so much from other observations or the lack of harmony between different parts or elements; incompleteness is referred to missing values; and timeliness is the degree to which data represent reality from the required point in time.This section presents the methods and the tools used for detecting them. Two methods for detecting outliers were used.

The first method for detecting outliers is the Local Outlier Factor (LOF). This method assigns to each item (e.g. a film) a degree of being an outlier based on a local density given by k nearest neighbors. The LOF depends on how isolated the item is with respect to the surrounding neighborhood[124]. This thesis executed the implementation of the LOF algorithm that is included in the *Weka* Software. This software receives a file with extension *.csv* or *.arff* and then executes the LOF algorithm with is located under Filters, Unsupervised, Attribute, LOF. After the execution of this algorithm, a new column entitled "LOF" is added to the LODMatrix, this column contains the value of the degree of outlier.

The second method for detecting outliers is the Tukey's method, which identify the outliers ranged above the 1.5 IQR (Interquartile Range). Listing D.1 shows the *R* code for detecting outliers into the LODMatrix, which is a modification of the code published on the web site of Dhana K [125]. This method showed the LODMatrix contained about 9400 outliers. In this code the variable *dt* contains the whole data for the LODMatrix, this variable contains also an extra column where films with value NA are outliers. These outliers are useful to detect films with erroneous or incomplete data.

R

```
#Packages
library(plyr)
library(dplyr)
library(ggplot2)
#Read CSV file
data<- read.csv(file="lodmatrixdb.csv",head=TRUE,sep=",")
# Remove empty Cells with ?
m <- as.matrix(data)
m[m=="?"] <- 0
df <- as.data.frame(m)
#Considering Name as categorical value
#Summarize other columns by counting length of
#each column for a specific Name
sData=ddply(df,~Name,summarise,country=length(Country),
        Starring=length(Starring),Directors=length(Directors),
        Producers=length(Producers),Writers=length(Writers),
        Year=length(Year),Subject=length(Subject))
head(sData)
dt=sData
var=sData[,4]
var_name <- eval(substitute(var),eval(dt))
#counts the number of NA values
na1 <- sum(is.na(var_name))
#It computes the mean value for var_name and eliminates
#the NA values before the computation
m1 <- mean(var_name, na.rm = T)
#With outliers
```

```
p <− ggplot(sData, aes(country, var_name))
p + labs(title = "With Outliers")+ geom_boxplot()
#The lower and upper "hinges" correspond to the first
#and third quartiles (the 25th and 75th percentiles).
qplot(var_name, geom="histogram")
#Detect the ouliers values: boxplot.stats()$out
#which use the Tukey's method to identify the outliers
#ranged above and below the 1.5∗IQR
outlier <− boxplot.stats(var_name)$out
mo <− mean(outlier)
var_name <− ifelse(var_name %in% outlier, NA, var_name)
#Without outliers
p <− ggplot(sData, aes(country, var_name))
p + labs(title = "Without Outliers")+ geom_boxplot()
qplot(var_name, geom="histogram")
na2 <− sum(is.na(var_name))
m2 <− mean(var_name, na.rm = T)
dt[as.character(substitute(var))] <− invisible(var_name)
row.has.na <− apply(dt, 1, function(x){any(is.na(x))})
#With Outliers Values
final.filtered.with <− dt[row.has.na,]
No_of_movie_with_outlier<−nrow(final.filtered.with )
#Without Outliers Values
final.filtered <− dt[!row.has.na,]
No_of_movie_without_outlier<−nrow(final.filtered.with)
# Name of the move
head(final.filtered[,c(1,1)])
nrow(sData)−nrow(final.filtered)
nrow(data)
```

Listing D.1. Outlier detection with Tukey's method

Other approach that may be useful for outliers detection is the Principal Component Analysis, a *R* script can be found at the web site of Shahram, A [126].

D.4 Selection of the K value for the K-Means algorithm

The selection of the K value was conducted measuring the values of the centroid distance, Gini distribution, and density with the software RapidMiner. The RapidMiner's process created for evaluating these measures is presented in Figures D.2 and D.3.

Figure D.2 shows the general view of the process. The operator *Retrieve lodmatrixdb* reads the file or the database containing the LODMatrixFull (see section D.2), then the operator *Select Attributes* filter the attributes to only attributes that are useful for the clustering. Next, the operator *Loop Parameters* is a operator that surround other operators that ares executed internally. Additionally, it is useful to set-up the execution, in this case it allows the experimentation to execute values of K in the range of [0,100] required for testing.

Figure D.2 shows the internal part of the process for evaluating the K value. The *Generate*

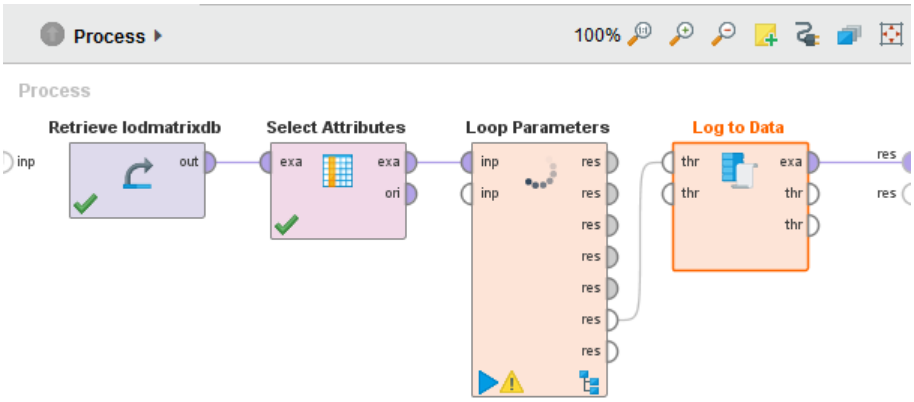


Figure D.2. Cluster Evaluation with K-Means - General View

ID operator adds a new attribute with id role in the LODMatrix. The operator *multiply* copies its inputs to all connected outputs, this is useful to distribute the execution into multiple paths.

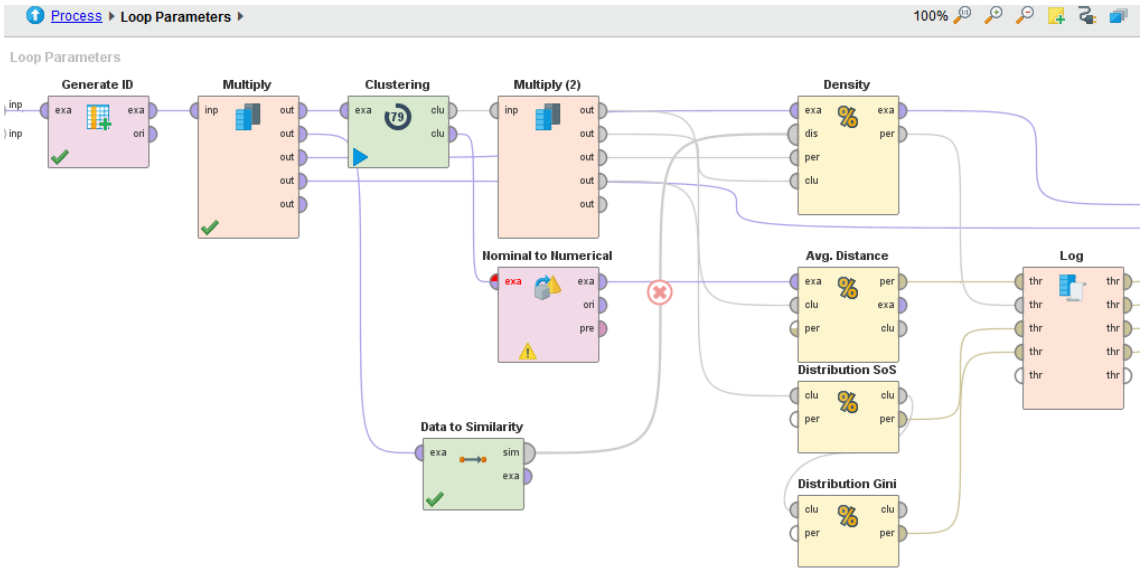


Figure D.3. Cluster Evaluation with K-Means - Internal View

The operator *Data to Similarity* measures the Mixed Euclidean similarity of each film of the LODMatrix with every other film of the same LODMatrix. This is necessary to compute the measure density. The *clustering operator* executes the k-means algorithm for the current k value as controlled by the *Loop Parameters* operator, the result of the k-means is then passed to the operators *Distance*, *Distribution SoS*, *Density*, and *Distribution Gini*.

The *Distance* operator computes the Average within centroid distance. This operator only support numerical data so before its execution data must be converted to numerical. In this process, this task is executed by the *Nominal to Numerical* operator. The *Distribution SoS* operator

measures the Sum Of Squares distance which is a distribution measure. The operator *Density* uses the output of the *Data to Similarity* and the clustering data to compute how many similar items where clustered in the same cluster. The *Distribution Gini* operator which measures the inequality among values of a frequency distribution. The results of all these operators are passed to the *Log* operator that creates a Log file with the results data. Finally, the operator *Log to Data* (Figure D.2) shows the results of the performance evaluation for the clustering in the range of K selected.

Appendix E

User Interfaces of the survey for evaluating the *AlLied* framework

This appendix shows examples of the graphic user interfaces used the user's survey for evaluating the framework *AlLied*. The application is available at `http://natasha.polito.it/RSEvaluation/`

Figure E.1 shows the home page of the evaluation survey, in this page the user evaluator enters his/her personal information Email address, age, gender, occupation and area of study.

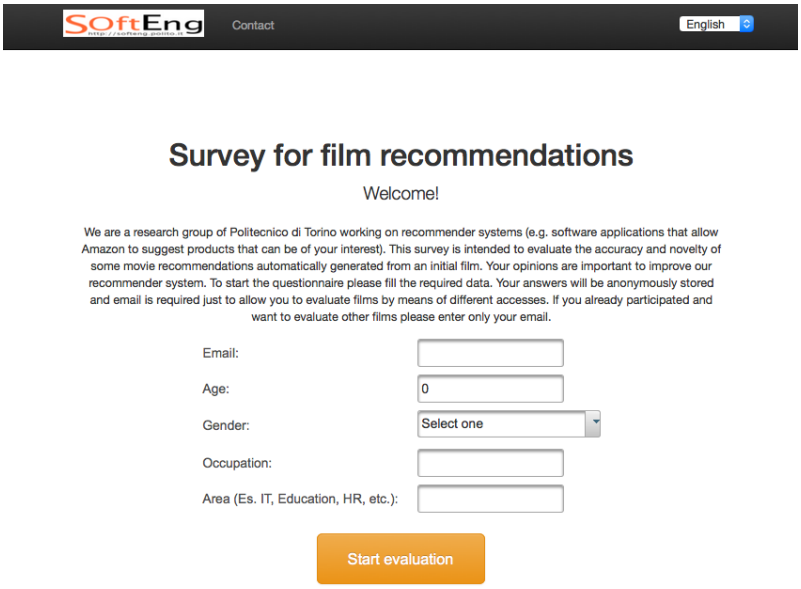


Figure E.1. Home page of the evaluation survey

Once the user has entered his/her personal information, the survey displays a list of films (query films) to be evaluated. Figure E.2 shows an example of these films.

Figure E.1 presents an example of the recommendations for a film selected in E.3, The user can select a film and asses if it is known for him/her, and if the user considers that this film is relevant for the query film.

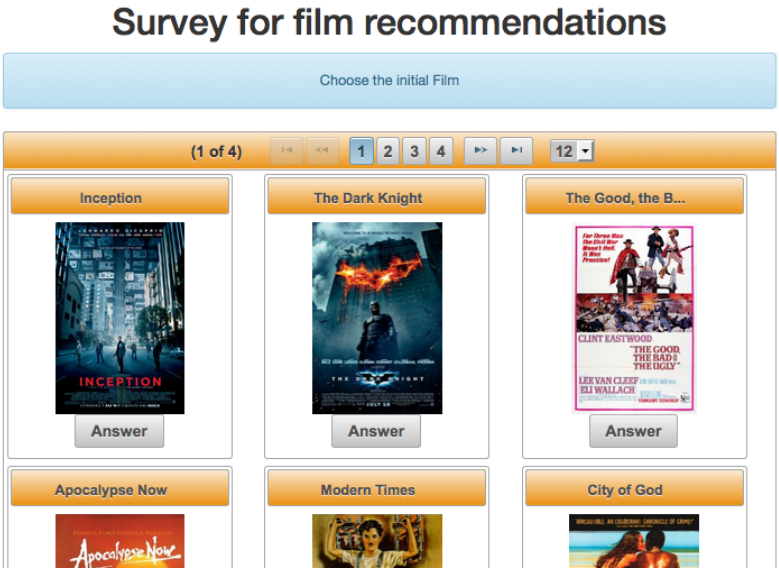


Figure E.2. Selecting a film for evaluation

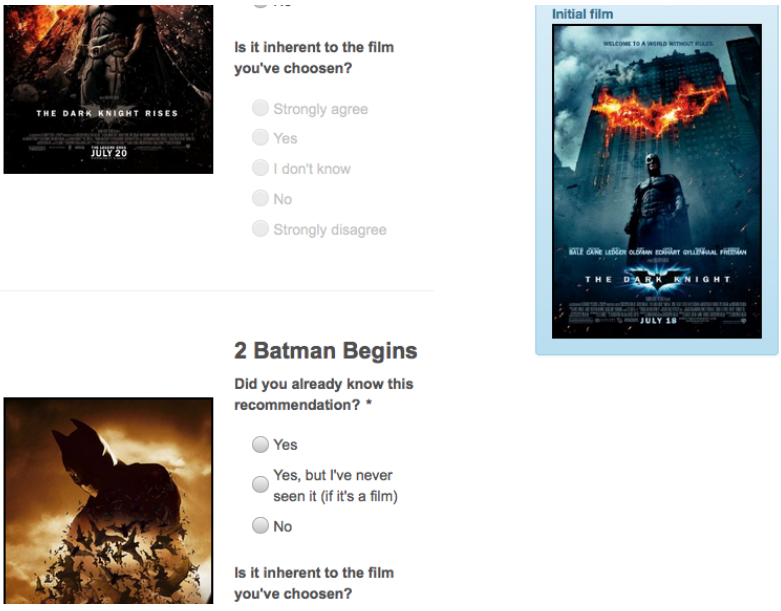


Figure E.3. Evaluating a film recommendations

Appendix F

Publications List

This appendix shows the list of research papers published during this PhD. First, publications which are closely related with the main topic of the PhD are listed, then other publications are listed[1].

F.1 Papers related with the main topic of this thesis

This section presents the list of papers related with RS based on Linked Data and their applications. The Quartile[2], Impact Factor (IF), and the category of the Publindex[3] are also shown.

F.1.1 Journal Papers

1. Figueroa, C., Vagliano, I., Rodríguez-Rocha, O., Torchiano, M., Faron-Zucker, C., Corrales, J.-C.,& Morisio, M. (2017). *Allied: A Framework for Executing Linked Data-based Recommendation Algorithms.* International Journal on Semantic Web and Information Systems, 13((3) 2017). ISSN: 1552-6283, **Quartile: Q1**, IF: 2.83, **Publindex: A2**.

2. Figueroa, C., Ordoñez, H., Corrales, J.-C., Cobos, C., Wives, L. K., & Herrera-Viedma, E. (2016). *Improving Business Process Retrieval Using Categorization and Multimodal Search.* Knowledge-Based Systems. ISSN: 0950-7051, **Quartile: Q1**, IF: 4.3, **Publindex: A1**.

3. Figueroa, C., Vagliano, I., Rodríguez Rocha, O.,& Morisio, M. (2015). *A systematic literature review of Linked Data-based recommender systems.* Concurrency and Computation:

---

[1]More information about my publications at: `https://www.researchgate.net/profile/Cristhian_Figueroa/contributions`

[2]The Impact Factor and the quartile data were retrieved from the SCImago Journal & Country Rank that is publicly available at `http://www.scimagojr.com`

[3]Publindex is a Colombian bibliographic index for rating and certifying scientific and technological publications. It is recognized by COLCIENCIAS in Colombia. Available at `http://publindex.colciencias.gov.co`

Practice and Experience, 27(17), 4659-4684. ISSN: 1532-0634 , **Quartile: Q2**, IF: 1.076, **Publindex: A2**.

4. Rodríguez Rocha, O., Vagliano, I., Figueroa, C., Cairo, F., Futia, G., Licciardi, C. A., Morando, F. (2015). *Semantic Annotation and Classification In Practice.* IT Professional, 17(12-IT-Enabled Business Innovation), 33-39. ISSN: 1520-9202, **Quartile: Q2**, IF: 1.067, **Publindex: A2**.

F.1.2 Conference Papers

1. 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.

2. Rodríguez Rocha, O., Figueroa, C., Vagliano, I.,& Moltchanov, B. (2014). *Linked Data-Driven Smart Spaces.* In S. Balandin, S. Andreev,& Y. Koucheryavy (Eds.), Internet of Things, Smart Spaces, and Next Generation Networks and Systems SE - 1 (Vol. 8638, pp. 3-15). Springer International Publishing. LNCS, ISSN: 0302-9743, **Publindex A2**.

F.2 Other papers published during the PhD

1. Ordóñez, A., Ordóñez, H., Figueroa, C., Cobos, C.,& Corrales, J. (2015). *Dynamic Reconfiguration of Composite Convergent Services Supported by Multimodal Search.* In W. Abramowicz (Ed.), Business Information Systems SE - 11 (Vol. 208, pp. 127-139). Springer International Publishing. LNBI, ISSN: 1865-1348, **Publindex A2**.

2. Figueroa, C., Corrales, C.,& Corrales, J. C. (2015). A Multilevel Approach for Business Process Retrieval. Revista Ingenierías Universidad de Medellín, 14(26), 177-190. **Publindex A2**.

3. Ordoñez, H., Figueroa, C., Corrales, J. C., Morisio, M., Cobos, C.,& Wives, L. K. (2014). *Business Process Indexing Based on Similarity of Execution Cases.* In 7th Euro American Conference on Telematics and Information Systems (p. 12:1–12:6). Valparaiso, Chile: **ACM**.

4. Tomassetti, F., Figueroa, C.,& Ratiu, D. (2014). *Tool-automation for supporting the DSL learning process.* In Second International Workshop on Open and Original Problems in Software Language Engineering (OOPSLE 2014). Antwerp, Belgium.

5. Rodríguez Rocha, O., Figueroa, C.,& Moltchanov, B. (2013). *A Subgraph Isomorphism Based Approach to Enable Discovery and Composition of Smart Space Elements.* In Internet of Things, Smart Spaces, and Next Generation Networking - ruSMART 2013 (pp. 84-93). San Petersburg. LNCS, ISSN: 0302-9743, **Publindex A2**.

6. Enríquez, G., Benavides, A., Ramirez, J. D., Figueroa, C.,& Corrales, J. C. (2012). *Control-Flow Patterns in Converged Services.* In SERVICE COMPUTATION 2012, The Fourth International Conferences on Advanced Service Computing. Niece, Francia.

7. Figueroa, C.,& Corrales, J. C. (2012). *Business Process Retrieval based on Behavioral Semantics.* Revista EIA, 105-120. **Publindex A2**.

8. *Book:* Figueroa, C., Corrales, J. C.,& Ramirez, G. A. (2012). *Recuperación Multinivel de Procesos de Negocio basada en Semántica del Comportamiento.* New York: Research and Innovation.