5-2017

Exploiting Semantic Distance in Linked Open Data for Recommendation

Sultan Dawood Alfarhood
University of Arkansas, Fayetteville

Exploiting Semantic Distance in Linked Open Data for Recommendation

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Computer Science

by

Sultan Alfarhood
King Saud University
Bachelor of Science in Computer Science, 2007
University of Arkansas
Master of Science in Computer Science, 2013

May 2017
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

Dr. Susan Gauch
Dissertation Director

Dr. Mark Arnold
Committee Member

Dr. Michael Gashler
Committee Member

Dr. Qinghua Li
Committee Member

Abstract

The use of Linked Open Data (LOD) has been explored in recommender systems in different ways, primarily through its graphical representation. The graph structure of LOD is utilized to measure inter-resource relatedness via their semantic distance in the graph. The intuition behind this approach is that the more connected resources are to each other, the more related they are. One drawback of this approach is that it treats all inter-resource connections identically rather than prioritizing links that may be more important in semantic relatedness calculations. Another drawback of current approaches is that they only consider resources that are connected directly or indirectly through an intermediate resource only. In this document, we show that different types of inter-resource links hold different values for relatedness calculations between resources, and we exploit this observation to introduce improved resource semantic relatedness measures that are more accurate than the current state of the art approaches. Moreover, we introduce an approach to propagate current semantic distance approaches that does not only expand the coverage of current approaches, it also increases their accuracy. To validate the effectiveness of our approaches, we conducted several experiments to identify the relatedness between musical artists in *DBpedia*, and they demonstrated that approaches that prioritize link types resulted in more accurate recommendation results. Also, propagating semantic distances beyond one hub resources does not only result in an improved accuracy, it also shows that propagating semantic distances beyond one hub resources improves the coverage of LOD-based recommender systems.

Acknowledgments

First and foremost, I would like to thank my advisor Dr. Susan Gauch, who has guided me to be a researcher! Without her guidance and persistent support, this dissertation would not have been possible.

Likewise, I express gratitude to my family for their love and support. It would be difficult to make it through without their support.

In addition, I acknowledge my sponsor, King Saud University, for giving me this opportunity. This scholarship gave me the opportunity to be challenged and made a big positive difference in my life.

Also, I thank my committee members, Dr. Mark Arnold, Dr. Michael Gashler and Dr. Qinghua Li, for the useful comments, remarks, and engagement through the learning process of this dissertation. As well, special thanks go out to the faculty and staff at the University of Arkansas for their commitment to the University and to the students.

Dedication

To my parents, Dawood and Haya.

Table of Contents

List of Figures

1 Introduction

Due to the massive amount of digital information available in recent years, it has become necessary to tailor the right information to the right user at the right time. Accordingly, new techniques and approaches have started to emerge that focus on matching information to users in order to help them make proper decisions. Some systems actively alert users to information or items that might be of interest to them; these methods are called recommender systems. Such systems have been embraced widely in various online platforms including commerce, news, and entertainment. There are numerous research works in this field that attempt to improve different aspects of recommender systems some of which are their recommendation accuracy, diversity, and novelty [1]. Researchers developing these systems continue to face several challenges, particularly a lack of *a priori* data needed in order for these systems to work appropriately. Several systems also lack sufficient semantic information about items, and semantic information about the relationships between items, so that related items can be accurately identified and recommended.

Information is widely available online through a different medium, particularly the world wide web (www). This information is available mainly as unstructured data as in the case of the text format that lacks sufficient information in order to effectively exploit the contents for advanced applications. The drive to address this issue has led to the creation of new standards and formats that enable consumption and distribution of structured data openly among different parties; this shareable structured data is known as *Linked Open Data (LOD)*. There are four principles of Linked Open Data [2]. Firstly, the Uniform Resource Identifier (URI) must be used to identify resources in any LOD dataset. Secondly, HTTP URIs must be used to look up resources. Thirdly, useful information must be provided on standard formats at each URI. Lastly, resources are linked for further exploration. Following this trend, various organizations have started to publish their data openly following LOD standards enabling organizations to interlink their concepts to other related concepts in different datasets. As a result, enormous datasets are now connected to each other, creating a huge map of datasets in different domains of knowledge. There are over 1014 linked data datasets in different domains [3]; some of these are specialized to a particular knowledge domain including music or books whereas others are generic, containing many cross-domain concepts such as the popular LOD provider, *DBpedia* [4]. Figure 1 displays a graph of several Linked Open Data datasets crawled in the year of 2014.

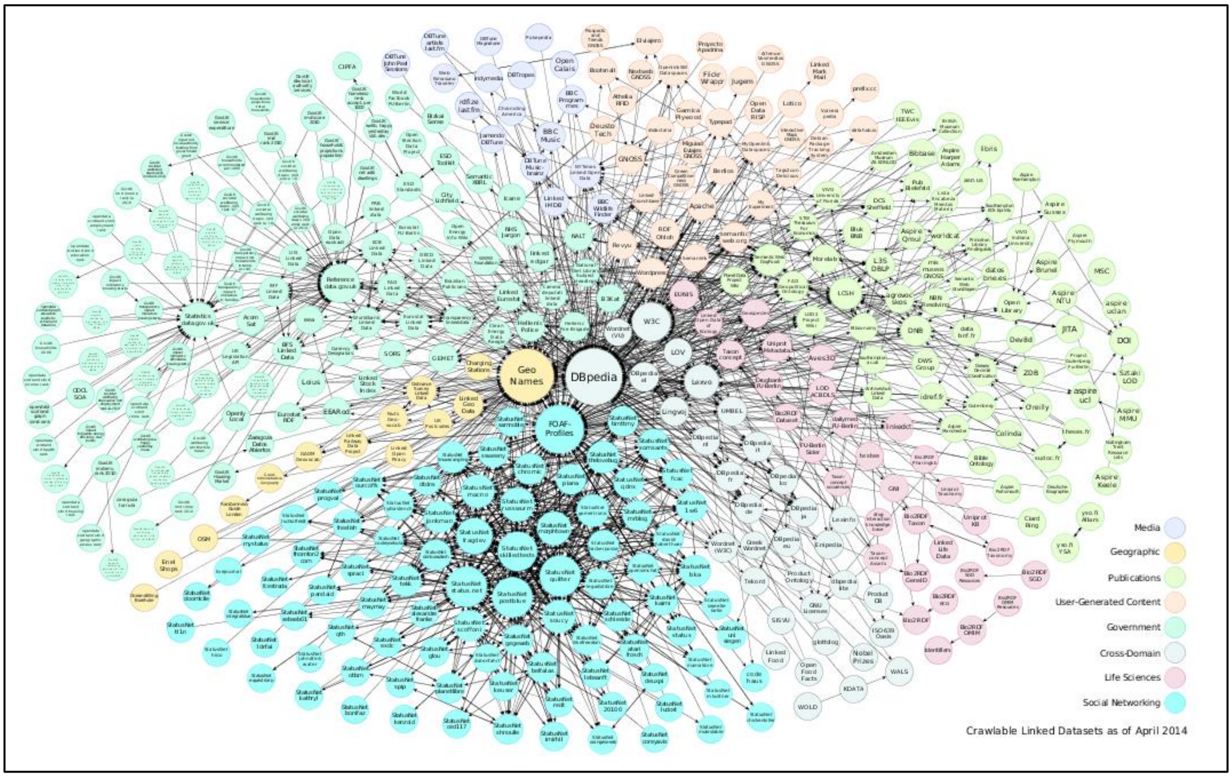


Figure 1: LOD cloud in 2014[1]

[1] Linked Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak. http://lod-cloud.net .

Because of its extensive offering of structured data in different domains, researchers have begun to investigate ways to exploit Linked Open Data in the field of recommender systems. One advantage for using LOD in recommender systems is that LOD provides broad open datasets containing multi-domain concepts with their relationships to each other, and these relations enable recommender systems to identify related concepts across collections [5]. Additionally, LOD standards and technologies ease the task of recommender systems by providing standard interfaces to retrieve required data, eliminating the need for additional computational processing of raw data. Furthermore, LOD provides ontological knowledge of the data that allows recommender systems to identify the relationship between concepts [6]. As a result, recommender systems can utilize LOD datasets and benefit from LOD's extensive open datasets to overcome the challenges presented by the lack of *a priori* data. LOD also facilitates explaining recommendation results of recommender systems since the relationship of items can be tracked easily in the LOD graph [7].

The use of LOD has been explored in recommender systems in different ways, primarily through exploiting its graph representation or through statistical approaches [8]. One approach that utilizes the graph structure of LOD in recommender system is to measure resources relatedness through their semantic distance in the graph [9] [10] [11]. The intuition behind this semantic distance approach is that the more connected resources are to each other in the LOD graph, the more related they are. This concept is the core of a resource relatedness measure, the *Linked Data Semantic Distance (LDSD)* [9], as well as a more recent measure based on it, *Resource Similarity (Resim)* [10]. These approaches analyze the connectivity between two resources, whether they are directly connected or indirectly connected through another resource, to generate a semantic distance value that represents the relatedness between the resources.

Resource similarity approaches are not only applicable to recommender systems; they can also be used in other applications as in the case of community detection in social networks [9]. One drawback of these approaches is that they treat all links between resources equally rather than prioritizing inter-resource links that may deliver additional value in semantic relatedness calculations. However, we argue in this document that different types of inter-resource links hold different importance for relatedness estimation. Furthermore, we exploit this observation to introduce improved resource semantic relatedness measures (*WLDSD*, *WTLDSD*, *WResim*, and *WTResim*) that extend current state of the art approaches. In addition, we present two new ways to calculate link weights based on probability theory (the *Resource-Specific Link Awareness Weights (RSLAW)*) and information theory (the *Information Theoretic Weights (ITW)*).

Another drawback of the existing approaches is that they only calculate the semantic distance between resources that are directly linked or indirectly linked through another resource. Thus, these approaches cannot calculate relatedness between resources if they are two resources away from each other in the LOD graph; they assume that these resources are not related. In this document, we propose a new approach that propagates semantic distances generated by current approaches to expand their coverage beyond current limitations.

To validate the effectiveness of our proposed approaches, we conducted several experiments to identify the relatedness between musical artists in *DBpedia* and we measured the recommendation accuracy based on the proposed approaches versus baselines based on the existing approaches (*LDSD* and *Resim*), and we found that several of our new approaches outperform the baselines. These experiments demonstrated that approaches that prioritize link types resulted in more accurate recommendation results. Also, the results show that the proposed propagated approach does not only increase the span of the semantic distance computations; it also increases the accuracy of the semantic distance calculations.

The contributions of this dissertation are the following:

1. Studying the significance of differentiating links types for relatedness purposes.
2. Proposing improved resource semantic relatedness approaches (*WLDSD*, *WTLDSD*, *WResim*, and *WTResim*) that are more accurate than the current state of the art.
3. Proposing two different ways to calculate links weights: *RSLAW* and *ITW*.
4. Proposing an approach that expands semantic distances generated by current approaches beyond current limitations.
5. Implementation and evaluating these approaches on top of *DBpedia*.

The remainder of this document is organized as follows: Section 2 presents related concepts and works followed by background information about the baselines that this document adopts and improves in Section 3. Next, Section 4 presents the design of the first goal of this document which details several approaches that exploit differential weights in LOD links for recommendation purposes. Subsequently, Section 5 details the second goal of this document, i.e., how current semantic distances can be propagated to expand their coverage. Afterward, the proposed system architecture is discussed in Section 6 followed by how the proposed approaches are evaluated against current state of art approaches in Section 7. Finally, Section 8 offers a summary of this document and presents some future work.

2 Literature Review

2.1 Recommender Systems

As information systems grow, an increasingly massive amount of digital information is available, and there is a need to tailor this information to the user when needed. One approach is to employ recommender systems, software methods and algorithms that suggest likely items of interest to users [12]. These systems identify the right information for users based on their needs. Recommender systems typically consist of three main components: background data, input data, and a recommendation algorithm [13]. First, there must be enough background data about the domain of the system such as the information about the items to be recommended and the relationships among them. Then, a sufficient information, input data, about the user is required to understand the user preferences in order to identify items related to his or her preferences. Information about the user is usually represented by a user profile that may be temporary or persistent between sessions. Finally, an appropriate algorithm is applied to the background data to suggests items to users based on their input data (user profiles).

Adomavicius and Tuzhilin [14] overview recommender systems and classify them into three general classes: content-based, collaborative, and hybrid. In content-based recommender systems, items are recommended to a user based on their similarities to other items the user has in his or her user profile. Collaborative recommender systems, on the other hand, recommend items to a user based on the similarity of the user and other users; then, items liked by their most similar users are recommended. Hybrid recommender systems combine different approaches, namely, content-based and collaborative methods into one system. Burke in [15] suggests two more classes: knowledge-based and demographic recommender systems. Knowledge-based recommender systems suggest items to users based on the inference of user's needs and preferences. Also, demographic recommender systems take the demographic profile of the user into consideration based on the assumption that people with different demographic niches have different needs. Some examples of these demographic niches are language, country, gender, and age. Still, demographic recommender systems have less research works in the literature compared to others types [12].

Focusing on one class, Lops et al. [16] present an overview of content-based recommender systems. They discuss some advantages of content-based recommender systems, some of which are their capability of handling new items, their user independence so they need only the user information to recommend items regardless of other users in the system. They are also transparent in terms of explaining how the recommended item was generated. On the other hand, content-based recommender systems face some challenges such as their ability to handle new users, their ability to produce diverse results, and their dependence on domain knowledge. Because they work by recommending similar items to previous items, they are only appropriate when there are some semantic features available about the items so that similarity between items can be detected, e.g., tagged documents or images.

Su and Khoshgoftaar [17] survey different collaborative filtering approaches. They start by discussing the challenges of using collaborative filtering approaches as in the case of the cold start problem for new items and users, the scalability challenge when systems grow to include huge information about items and users. In addition, collaborative systems suffer from the synonymy problem for items with different names, shilling attacks wherein malicious users can affect the system bias to some content, and privacy concerns surrounding users' information. Nonetheless, because they work by exploiting users with similar patterns of preferences, they can be used to recommend items that have little or no semantic information available. Su and

Khoshgoftaar categorize collaborative techniques into three classes: memory-based, model-based and hybrid. Memory based systems estimate the similarity between users or items based on the user rating data in order produce recommendations. Model-based systems, on the other hand, utilize the rating data to learn a model that will be used to make the recommendation. Hybrid approaches combine both memory-based and model-based techniques in order achieve better recommendation results.

Parra and Sahebi [18] also provide an overview of recommender systems, and they discuss their sources of knowledge such as users' ratings, implicit feedback and interaction in the system, social tags and keywords, online social networks, and contexts, namely, location and time. They also discuss evaluation metrics used in recommender systems including prediction-based metrics which compare algorithms based on their ability to make fewer mistakes in predicting recommended items. *Mean Absolute Error (MAE)*, *Mean Squared Error (MSE)*, and *Root Mean Squared Error (RMSE)* are some examples of prediction-based metrics. Other metrics are related to the information retrieval field wherein users are provided with a list of recommended items and they classify these results as relevant or not. Examples of information retrieval metrics are *Precision*, *Mean Average Precision (MAP)*, *Recall*, and *Discounted Cumulative Gain (DCG)*. Additionally, recommender systems can be evaluated based on their diversity, novelty, and coverage.

The field of cross-domain recommender systems has been explored lately by the recommender systems community due to its promise to allow progress on several problems of recommender systems contributing to the cold start problem and producing better recommendation results. The cross-domain recommendation problem has been formally defined by Cremonesi et al. [1] as suggesting new and unknown items in a target domain to the users of a source domain where the preferences of the users exist. They also introduced a collaborative filtering based cross-domain recommender system that relies on a modified standard neighborhood-based collaborative filtering method in this paper. Furthermore, Fernández-Tobías et al. [19] have introduced a similar formal definition of the cross-domain recommendation problem, and they characterized several cross-domain related strategies that were proposed in the literature. Cantador and Cremonesi came together later in [20] and characterized recommender systems domains into four levels: item attribute, item type, item, and system level.

2.2 Linked Open Data (LOD)

Linked Data is a term that describes a model of published data that follows four rules [2]:

(1) using *Uniform Resource Identifiers (URI)* to identify things (resources).

(2) using HTTP URIs in order look up resources.

(3) providing useful information at these URIs based on standard formats (e.g., RDF, SPARQL).

(4) connecting to other resources to allow for further exploration.

Additionally, data should be available on the web and be under an open license in order to be fully qualified as *Linked Open Data (LOD)* [21]. Linked Open Data requires standard formats to distribute and consume data, including *Uniform Resource Identifiers (URI),* which is a set of characters used to identify a resource following the *Internet Engineering Task Force (IETF)* standard (RFC 3986) [22]. Furthermore, the *Resource Description Framework (RDF)* standard is an XML-based format used to specify the meaning of links between resources. SPARQL, *SPARQL Protocol and RDF Query Language,* is a query language used to query and manipulate data stored as RDFs [23] [24].

Schmachtenberg et al. [3] analyzed the LOD cloud datasets to study their growth using a Linked Data crawler. They found that the number of LOD datasets has increased from 294 in 2011 to 1091 datasets in 2014. They also observed that 77% of LOD datasets utilize well-known vocabularies such as *Friend of a Friend*[2] *(FOAF)*, which is an ontology to describe people, while the usage of proprietary vocabularies has declined from 64.41% in 2011 to 23.08% in 2014.

One of the most popular Linked Open Data providers is *DBpedia*, which Auer et al. [4] describe as a community project to extract structured data from *Wikipedia* and make it available openly on the web. They detail the extraction of the *DBpedia* dataset into two steps: (1) they map already available structured data directly into RDF format, and (2) they extract additional useful information from article texts and then make it available in RDF format. *DBpedia* is interconnected with several LOD providers including *WordNet*, *MusicBrainz, US Census, Geonames*, the *DBLP bibliography*, and others. In addition, *Freebase* [25], originally introduced in [26], is another Linked Open Data provider of human knowledge with a large community-based data in diverse domains. *Freebase* consists of over 4000 resource classes with more than 125,000,000 links. It is currently owned by *Google*, which used it among others to build its own knowledge graph [27].

Since LOD providers utilize different ontologies, Jain et al. [28] presented a system, called *BLOOMS*, that align LOD datasets ontologies even if they are not directly linked. This system matches LOD ontologies with the *Wikipedia* hierarchy to link these ontologies. After preprocessing each ontology, this system constructs a concept tree per ontology though matching it with *Wikipedia* articles. These concept trees are constructed using the *Wikipedia* categories. After that, the concept trees are compared and aligned to each other.

[2] www.foaf-project.org

In one LOD real life application, Kobilarov et al. [29] show how BBC uses several LOD providers to integrate data and link documents together. They exploit *DBpedia* to link all their published programs data and documents. They also built a web musical portal that links music to their programs on top of the music-specific LOD provider, *MusicBrainz*.

2.3 LOD in Recommender Systems

Figueroa et al. [8] review recommender systems that utilize linked data for recommendation purposes. They start by discussing the motivation behind adopting LOD in recommender systems, and one of the most popular motivation is the lack of semantic information about items to be recommended. In addition, using LOD in recommender systems can help to solve the cold-start problem especially in collaborative-filtering systems. Figueroa et al. group algorithms in this field into two classes: graph-based algorithms, and statistical information techniques. Graph-based algorithms take the graph nature of LOD into consideration by working directly in the LOD graph to find linked related items. In contrast, statistical approaches extract content features from the LOD graph and then apply a recommendation algorithm to these features.

Di Noia and Ostuni [6] also present an overview of recommender systems generally and follow it by discussing how LOD can be employed to build semantics-aware recommender systems. They suggest that there are two essential components for LOD-based recommender systems to work properly: an item linker and an item graph analyzer. The item linker component is responsible for mapping items in the system with the corresponding item in the LOD dataset. The item graph analyzer generates a subgraph of items related to the item after analyzing the relationship between this item and other items in the LOD graph.

Passant [9] suggests an approach to exploiting LOD in recommender systems by computing the semantic distance between resources in the LOD. His approach, called *Linked*

Data Semantic Distance (LDSD), exploits direct links between resources along with indirect resources through an intermediate resource to calculate a semantic distance between these resources. Utilizing this approach, Passant in [30] has created a music recommender system, called *dbrec*, that is built on top of the popular LOD provider, *DBpedia*, in order to recommend musical artists and bands. This system starts by reducing the LOD dataset to a compact one that enables efficient semantic distance computations. It then calculates the semantic distance between each pair of musical artists or bands. Finally, utilizing these semantic distances, related artists are generated for the user. Exploiting the aforementioned concept, Piao et al. [10] introduced an improved linked data semantic distance approach, called *Resource Similarity* (*Resim*), that revised the original *LDSD* approach overcoming some its weaknesses namely equal self-similarity, symmetry, and minimality issues. They also improved their approach in [11] by applying different normalization methods based on the path appearances in the LOD graph. One drawback of these approaches is that they handle all resources connections equally and do not prioritize resources links that hold additional value in semantic relatedness calculations. Also, they can only calculate the semantic distance between two directly connected resources or indirectly connected through an intermediate resource only. In a similar approach, Leal et al. [31] [32] present another semantic relatedness approach, called *Shakti*, that measures the relatedness between LOD resources. In this approach, the relatedness between resources is measured based on their proximity. In particular, the proximity is measured based on the number of indirect links penalized by their distance length. Still, *LDSD* and *Resim* accuracy outperform *Shakti* as demonstrated by [10].

Rather than basing resource similarity on specific links and link types between pairs of resources, Nguyen et al. [33] investigate the usage of two structural context similarity approaches of graphs in the field of LOD recommender systems. They found that two metrics *SimRank* and *PageRank*, are promising in this field and can produce some novel recommendations, but they carry a high-performance cost. Furthermore, Damljanovic et al. [5] present a concept recommender system based on LOD that assists users choosing proper concept tags and topics to improve their web search experience. They introduced a similarity-based approach relying on the relationship between concepts in the LOD graph. They also present another statistical-based method to calculate concept similarities and a comparison of both approaches to the *Google Adwords Keyword Tool*. They conclude that the graph-based method outperforms their baseline in relatedness measures while the statistical method came up with better-unexpected results. Correspondingly, Fernández-Tobías et al. [34] have developed a cross-domain recommender system that relies on LOD to link concepts from two different domains. They extract information about the two domains from LOD sources and then link concepts using a graph-based distance between these concepts. Based on this approach, they developed in [35] a recommender system for the domains of architecture and music to suggest musical artists based on a selected location built on top of *DBpedia*.

Di Noia et al. [36] show that LOD has the potential to be effectively used in content-based recommender systems, particularly because LOD can help overcome issues of items that are described by limited content. They also describe [37] a content-based recommender system that employs LOD datasets, for instance, *DBpedia*, *Freebase,* and *LinkedMDB* to recommend movies. They utilize these LOD datasets to gather contextual information about movies such as actors, directors, and genres and then apply a content-based recommendation approach to generate recommendation results. Similarity, Ostuni et al. [38] presented a location-based movie recommender system, called *Cinemappy*, that takes into consideration both time and location of the user to produce a recommendation. This content-based recommender system leverages *DBpedia* to determine movies similarities based on their connectivity, i.e., exploiting linkages between movies. In addition, Ostuni et al. [39] present a hybrid LOD-based recommender system that exploits users' implicit feedback and is built on top of *DBpedia*. Semantic information about items in the user profile and items in *DBpedia* are merged into a unified graph from which path-based features are extracted for the recommendation algorithm. Similarly, Ostuni et al. [40] also introduce a content-based recommender system that generates semantic item similarities using *DBpedia*. The semantic similarity between items is calculated using a neighborhood-based graph kernel that finds local neighborhoods of these items. Later, Nguyen et al. [41] examine whether or not LOD providers can improve recommender systems in terms of precision, diversity, and novelty. They evaluated four different recommendation approaches employing the LOD providers *DBpedia* and *Freebase* in the music domain. They argued that using *DBpedia* enhances novelty of the recommendation results whereas using *Freebase* increases the coverage of the recommender system.

Meymandpour and Davis [42] describe a LOD-based recommender system that combines semantic analysis of items with collaborative filtering approaches to overcome the item cold-start problem. They found that their semantic approach works well when combined with collaborative filtering methods to improve recommendations. The collaborative filtering is particularly helpful when there is limited information available in the user profile. Likewise, Heitmann and Hayes [43] exploit LOD to overcome common collaborative-filtering challenges as in the case of the new-user, new-item, and sparsity problems. Heitmann [44] has also developed an open framework for cross-domain personalization relying on the data representation in LOD. The

LOD representation is used to model the user profile as well as the system catalog of items that results in an open framework for recommender systems.

In other work, Musto et al. [45] investigated the potential contribution of LOD to recommender systems by evaluating how features extracted from LOD can affect the accuracy of different recommendation algorithms. They found that recommendation approaches with features extracted from LOD outperformed non-LOD-based approaches. Similarly, Peska and Vojtas [46] [47] show that LOD can be used effectively to enhance recommender systems in current e-commerce sites. They rely on LOD sources to fetch additional information about items in current systems in order for content-based recommender systems to work properly. In addition, Kabutoya et al. [48] propose a hybrid movie recommender system that combines content-based and collaborative filtering techniques. Their system obtains movies' metadata from a LOD provider, *MovieLens*, and then applies a collaborative-based technique to tackle the cold start problem.

Clearly, there is a very active research community focusing on applying LOD sources to recommender systems. Our work builds on these projects but differs in that it takes the advantage of the LOD nature to improve current relatedness measures approaches through prioritizing some links that hold more relatedness value between the LOD resources. It also expands semantic distance generated by current approaches to include additional resources beyond current limitations.

3 Background

Linked Open Data is used in the field of recommender systems in different ways. Its semantic structured data can be exploited to improve recommender systems, particularly content-based systems. In one approach, Passant [9] introduced a semantic distance-based approach within LOD to identify related resources within recommender systems. It measures the relatedness between resources in LOD by calculating a semantic distance between them such that resources are considered more related (closer) if they are connected to each other through several paths.

Primarily, Linked Open Data is designed as resources (nodes) connected semantically to each other via links (edges) as in a graph. This graph-based nature is essential to formally define LOD instances. This document adopts the same definition for LOD datasets as the one described in [9]:

A Linked Open Data dataset is a graph G such as $G = (R, L, I)$ in which:

$R=\{r_1, r_2, \dots , r_m\}$ is a set of resources identified by their URI (Unique universal identifier)

$L = \{l_1, l_2, \dots , l_n\}$ is a set of typed links identified by their URI

$I = \{i_1, i_2, \dots , i_o\}$ is a set of instances of these links between resources, such as $i_i = <l_j, r_a, r_b>$

To put this definition in perspective, a simple graph instance is shown in Figure 2. Part A of the chart is a generic version that follows the definition where $R= \{r_1, r_2, r_3, r_4, r_5\}$, $L = \{l_1, l_2, l_3, l_4\}$, and $I = \{<l_1, r_1, r_2>, <l_1, r_3, r_2>, <l_2, r_4, r_2>, <l_3, r_5, r_3>, <l_3, r_5, r_4>, <l_4, r_3, r_2>\}$. The same example can be understood better by applying it in the music domain where the resources can be artists, songs, etc. and the links shows the relationship between these resources as in part B of Figure 2.

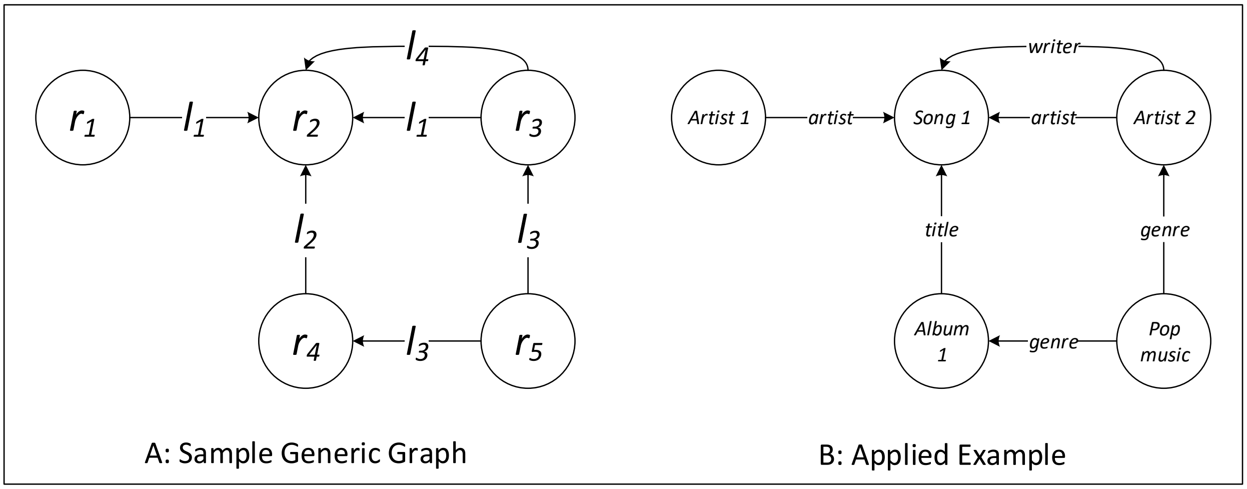


Figure 2: Sample graph

3.1 Direct Connectivity (*DC*)

Connectivity between resources in the graph can show relatedness, and the more connected the resources the more relatedness indication there is. In this context, a direct connection between two resources exists when there is a distinct direct link (directional edge) between these two. The *Direct Connectivity* (*DC)* can be calculated as the total number of distinct direct links between two resources. Formally, *Direct Connectivity* (*DC*) between two resources r_a and r_b is the sum of *Direct Link Connectivity* (*DLC*) over all links that connect them and originated from r_a as follows:

$$DC(r_a, r_b) = \sum_i DLC(l_i, r_a, r_b) \ , \qquad \{\forall\, l_i |\ \exists\langle l_i, r_a, r_b\rangle\}$$

The *Direct Link Connectivity* (*DLC*) between two resources r_a and r_b through a link of type l_i is equal to one if there a link of type l_i exists that connects the resource r_a to the resource r_b as follows:

$$DLC(l_k, r_a, r_b) = \begin{cases} 1 & \text{if the link } \langle l_k, r_a, r_b \rangle \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

By looking at the example shown in Figure 2, the direct connectivity between r_3 and r_2 is two ($DC(r_3, r_2) = 2$) because they are connected by l_1 and l_4, and the direct connectivity between r_2 and r_3 is zero ($DC(r_2, r_3) = 0$) as there are no direct links originating from r_2.

3.2 Indirect Connectivity (*IC*)

Resources can also be indirectly connected to other resources. Indirect connectivity between two resources occurs when they are connected through another resource, and these connections are either both incoming or both outgoing through the intermediate resource. Therefore, there are two types of indirect connections: incoming and outgoing. An incoming indirect connection between two resources r_a and r_b exists if there is a resource r_c such that r_c is directly connected to both r_a and r_b as in part A of Figure 3. Likewise, an outgoing indirect connection between two resources r_a and r_b exists if there is a resource r_c such that both r_a and r_b are directly connected to r_c as in part B of Figure 3.

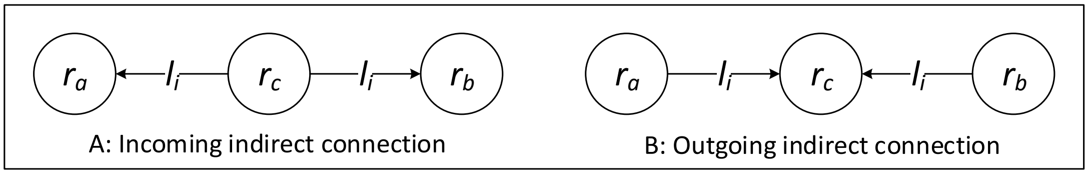

r_a
l_i
r_c
l_i
r_b
A: Incoming indirect connection
r_a
l_i
r_c
l_i
r_b
B: Outgoing indirect connection

Figure 3: Indirect connection types

Formally, the *Incoming Indirect Connectivity* (*IC_i*) between two resources r_a and r_b is the sum of the *Incoming Indirect Link Connectivity* (*ILC_i*) of all links that connect them as follows:

$$IC_i(r_a, r_b) = \sum_n \sum_j ILC_i(l_j, r_n, r_a, r_b)$$

The *Incoming Indirect Link Connectivity* (*ILC_i*) between two resources r_a and r_b is equal to one if there is a resource r_c such that r_c is directly connected to both r_a and r_b via a link of type l_k as follows:

$$ILC_i(l_k, r_c, r_a, r_b) = \begin{cases} 1 & \{\exists\, r_c \mid \langle l_k, r_c, r_a\rangle \& \langle l_k, r_c, r_b\rangle\} \\ 0 & \text{otherwise} \end{cases}$$

Likewise, the *Outgoing Indirect Connectivity* (*IC_o)* between two resources r_a and r_b is the sum of the *Outgoing Indirect Link Connectivity* (*ILC_o)* of all links that connect them as follows:

$$IC_o(r_a, r_b) = \sum_n \sum_j ILC_o(l_j, r_n, r_a, r_b)$$

The *Outgoing Indirect Link Connectivity* (*ILC_o*) between two resources r_a and r_b is equal to one if there is a resource r_c such that both r_a and r_b are directly connected to r_c via a link of type l_k as follows:

$$ILC_o(l_k, r_c, r_a, r_b) = \begin{cases} 1 & \{\exists\, r_c \mid \langle l_k, r_a, r_c\rangle \& \langle l_k, r_b, r_c\rangle\} \\ 0 & \text{otherwise} \end{cases}$$

Following the example in Figure 2, the incoming indirect connectivity between r_3 and r_4 is one ($IC_i(r_3, r_4) = 1$) through the resource r_5 linked by the link type l_3, however, the outgoing indirect connectivity between r_3 and r_4 is zero ($IC_o(r_3, r_4) = 0$) since there is no resource such that both r_3 and r_4 are directly connected to through the same link type.

The *Indirect Link Connectivity* (*ILC*) notation can be generalized for all intermediate resources as follows[3]:

$$ILC_i(l_k, r_a, r_b) = \sum_n ILC_i(l_k, r_n, r_a, r_b)$$

[3] These versions of *ILC* accept three inputs instead of four as in the regular *ILC*

$$ILC_o(l_k, r_a, r_b) = \sum_n ILC_o(l_k, r_n, r_a, r_b)$$

3.3 Linked Data Semantic Distance (*LDSD*)

Based on the previously mentioned concepts, Passant [9] defines an approach that measures the relatedness between two resources in the LOD using direct connections only. This metric is called the *Linked Data Semantic Distance - direct* ($LDSD_d$); it is essentially the inverse of the direct connectivity between the two resources. Since the links in LOD are directional, the formula includes a component for links from r_a to r_b and vice versa. The $LDSD_d$ is calculated as follows:

$$LDSD_d(r_a, r_b) = \frac{1}{1 + DC(r_a, r_b) + DC(r_b, r_a)}$$

Continuing the example from the previous section:

$$LDSD_d(r_1, r_2) = \frac{1}{1 + DC(r_1, r_2) + DC(r_2, r_1)} = \frac{1}{1 + 1 + 0} = 0.5$$

$$LDSD_d(r_3, r_2) = \frac{1}{1 + DC(r_3, r_2) + DC(r_2, r_3)} = \frac{1}{1 + 2 + 0} = 0.33$$

In this example, the value of $LDSD_d(r_3, r_2)$ is smaller than $LDSD_d(r_1, r_2)$ which indicates r_3 is closer to r_2 than r_1 when calculated using direct connections only.

In the same fashion, Passant [9] defines another metric called the *Linked Data Semantic Distance - indirect* ($LDSD_i$) based on the *Indirect Connectivity* concept. It is essentially the inverse of both incoming and outgoing indirect connectivity between the two resources as follows:

$$LDSD_i(r_a, r_b) = \frac{1}{1 + IC_i(r_a, r_b) + IC_o(r_a, r_b)}$$

Meanwhile, the indirect connectivity is bidirectional by its nature ($ID_i(r_a, r_b)$ = $ID_i(r_b, r_a)$), and there is no need to include the IC_i or IC_o twice, once per direction, as in the $LDSD_d$. Passant also [9] evaluated different combinations of the direct and indirect connectivity measures and found that the best performing formula that measures the relatedness between two resources in LOD, called *Linked Data Semantic Distance - combined normalized* ($LDSD_{cn}$), is the following[4]:

$$LDSD_{cn}(r_a, r_b) = \frac{1}{1 + DC'(r_a, r_b) + DC'(r_b, r_a) + IC'_i(r_a, r_b) + IC'_o(r_a, r_b)}$$

where $DC'(r_a, r_b)$ is merely the direct connectivity (*DC*) between resources r_a and r_b normalized by the log of all outgoing links from the resource r_a as follows:

$$DC'(r_a, r_b) = \sum_j \frac{DLC(l_j, r_a, r_b)}{1 + \log(\sum_n DLC\left(l_j, r_a, r_n\right))}$$

$DC'(r_b, r_a)$ is the direct connectivity (*DC*) between resources r_b and r_a normalized by the log of all outgoing links from the resource r_b as follows:

$$DC'(r_b, r_a) = \sum_j \frac{DLC(l_j, r_b, r_a)}{1 + \log(\sum_n DLC\left(l_j, r_b, r_n\right))}$$

$IC'_i(r_a, r_b)$ is the incoming indirect connectivity (IC_i) between resources r_b and r_a normalized by the log of all incoming indirect links to the resource r_a as follows:

$$IC'_i(r_a, r_b) = \sum_j \frac{ILC_i(l_j, r_a, r_b)}{1 + \log(\sum_n ILC_i\left(l_j, r_a, r_n\right))}$$

[4] The definition of LDSD is rewritten to be consistent with this document's concepts

$IC'_o(r_a, r_b)$ is the outgoing indirect connectivity (*IC_o*) between resources r_b and r_a normalized by the log of all outgoing indirect links from the resource r_a as follows:

$$IC'_o(r_a, r_b) = \sum_j \frac{ILC_o(l_j, r_a, r_b)}{1 + \log(\sum_n ILC_o\left(l_j, r_a, r_n\right))}$$

This algorithm incorporates both direct and indirect connectivity between two resources in both ways and normalizes the semantic distance value based on the count of each link instances to give less value for most used links regardless of thier importance for recommendation purposes. The semantic distances generated by the $LDSD_{cn}$ approach are ranged from zero to one; zero represents that the two resources are 100% related while the value one represents no relatedness at all between them. Since $LDSD_{cn}$ utilizes both the direct connectivity (*DC*) and the indirect connectivity (IC_i and IC_o) only to calculate the semantic distance, it can only compute the semantic distance between two directly linked resources or indirectly linked through an intermediate resource only. As a result, resources that are located more than one resource away are automatically considered unrelated to each other. The $LDSD_{cn}$ approach is our first baseline in this document in which we refer to it as just *LDSD* for simplicity. We also discuss a second baseline in the next section.

3.4 Resource Similarity (*Resim*)

Resource Similarity (*Resim*) [10] is an improved linked data semantic distance approach that enhances the original *LDSD* approach by overcoming some of its weaknesses, namely, equal self-similarity, minimality, and symmetry issues. In *LDSD*, the semantic distance between each resource and itself can vary between resources $(\text{i.e.}, LDSD(r_a, r_a) = 0.2 \; and \; LDSD(r_b, r_b) = 0.4)$, which violates the equal self-similarity property that is desirable for similarity measures since $(LDSD(r_a, r_a) \neq \; LDSD(r_b, r_b))$. Moreover, a semantic distance between each resource and itself does not always equal zero in *LDSD* ($LDSD(r_a, r_a) \neq 0$), which violates the minimality property. *Resim* solves these issues by including a criterion that ensures the semantic distance between each resource and itself is always zero. In addition, a semantic distance between two resources r_a and r_b is not always equal to the semantic distance between r_b and r_a ($LDSD(r_a, r_b) \neq LDSD(r_b, r_a)$) because the normalization in *LDSD* is performed to one resource only; hence, there is no symmetry. *Resim* solves this issue by using a consistent normalization method that depends on shared properties between resources. The *Resim* measure solves these issues as follows[5]:

$$Resim(r_a, r_b) = \begin{cases} 0 & if\ URI(r_a) = URI(r_b)\ or\ r_a\ owl{:}sameAs\ r_b \\ LDSD_\gamma(r_a, r_b) & if\ LDSD_\gamma(r_a, r_b) \neq 1 \\ Property_{sim}(r_a, r_b) & otherwise \end{cases}$$

The *Linked Data Semantic Distance* ($LDSD_\gamma$) component is calculated as follows:

$$LDSD_\gamma(r_a, r_b) = \frac{1}{1 + RC(r_a, r_b) + RC(r_b, r_a) + RI_i(r_a, r_b) + RI_o(r_a, r_b)}$$

where $RC(r_a, r_b)$ is the direct connectivity (*DC*) between resources r_a and r_b normalized by the log of number of instances of a link l_j as follows:

$$RC(r_a, r_b) = \sum_j \frac{DLC(l_j, r_a, r_b)}{1 + log(\sum_m \sum_n DLC(l_j, r_m, r_n))}$$

$RC(r_b, r_a)$ is the direct connectivity (*DC*) between resources r_b and r_a normalized by the log of number of instances of a link l_j as follows:

$$RC(r_b, r_a) = \sum_j \frac{DLC(l_j, r_b, r_a)}{1 + log(\sum_m \sum_n DLC(l_j, r_m, r_n))}$$

[5] The definition of *Resim* is rewritten to be consistent with this document’s concepts

$RI_i(r_a, r_b)$ is the incoming indirect connectivity (*IC_i*) between resources *r_a* and *r_b* through a resource *r_j* normalized by the log of all incoming indirect links to the resource *r_j* with a link type of *l_i* as follows:

$$RI_i(r_a, r_b) = \sum_i \sum_j \frac{ILC_i(l_i, r_j, r_a, r_b)}{1 + log(\sum_n ILC_i(l_i, r_j, r_n))}$$

$RI_o(r_a, r_b)$ is the outgoing indirect connectivity (*IC_o*) between resources *r_a* and *r_b* through a resource *r_j* normalized by the log of all outgoing indirect links from the resource *r_j* with a link type of *l_i* as follows:

$$RI_o(r_a, r_b) = \sum_i \sum_j \frac{ILC_o(l_i, r_j, r_a, r_b)}{1 + \log(\sum_n ILC_o(l_i, r_j, r_n))}$$

In addition, $Property_{sim}$ calculates the similarity of shared links types between resources *r_a* and *r_b* if the semantic distance generated by *LDSD* is one as follows:

$$Property_{sim}(r_a, r_b) = 1 - \left(\frac{\sum_i \left(\frac{C_{sip}(l_i, r_a, r_b)}{\sum_m \sum_n DLC(l_i, r_m, r_n)} \right)}{C_{ip}(r_a) + C_{ip}(r_b)} + \frac{\sum_i \left(\frac{C_{sop}(l_i, r_a, r_b)}{\sum_m \sum_n DLC(l_i, r_m, r_n)} \right)}{C_{op}(r_a) + C_{op}(r_b)} \right)$$

where:

- $C_{sip}(l_i, r_a, r_b)$ is the number of shared incoming links of type *l_i* between resources *r_a* and *r_b*
- $C_{sop}(l_i, r_a, r_b)$ is the number of shared outgoing links of type *l_i* between resources *r_a* and *r_b*
- $\sum_m \sum_n DLC(l_i, r_m, r_n)$ represents the number of instances of the link type *l_i*
- $C_{ip}(r_a)$ is the total number of incoming links to a resource *r_a*
- $C_{op}(r_a)$ is the total number of outgoing links from a resource *r_a*

The $Property_{sim}$ estimates the similarity of shared incoming and outgoing link types by calculating the ratio between the number of shared link types among the two resources and the total number of link types in the dataset. This ratio is normalized by the total number of incoming and outgoing links to the resources. It is another improvement of the *Resim* approach over *LDSD*, and it is useful when the semantic distance generated by the *LDSD* component is one ($LDSD_{\gamma}(r_a, r_b) = 1$); a case indicating that there is no relatedness between r_a and r_b or there is no direct or indirect links between the resources.

The *Resim* approach is the second baseline in this document. Our proposed enhancements are discussed in the following sections.

4 Exploiting Differential Weights in LOD Links for Recommendation Purposes

Linked Open Data is used in the field of recommender systems in different ways. Their structured semantic data can be exploited to improve recommender systems, particularly content-based systems. Some approaches [9] [10] incorporates semantic distance-based approaches within LOD to identify related resources within recommender systems. They measure the relatedness between resources in LOD by calculating a semantic distance between them such that resources are considered more related (closer) if they are linked to each other through several paths. One drawback of these approach is that all links (paths) in LOD are treated equally, and there is no distinction between links that have no significant impact on recommendations and those that should influence recommendations.

The first goal of this document investigates this case and suggests several approaches to address it. First, we introduce weighted variations of our baselines (*WLDSD* and *WResim*) to assess the significance of prioritizing some link paths in LOD for recommender systems and then propose two different approaches to calculate link weights (*RSLAW* and *ITW*). Then, we study the significance of recognizing link types by introducing typeless variations of our baselines (*TLDSD* and *TResim*). Lastly, we combine these two approaches to introduce weighted typeless variations of the baselines (*WTLDSD* and *WTResim*) to evaluate the effects of prioritizing some link paths in LOD regardless of their type.

4.1 Weighted Semantic Distance

Links between different resources in LOD can be of different importance for recommender systems. Recognizing these differences could be vital in order to produce better recommendation results. For instance, singers who create a joint work (duet) together are likely more related to each other than singers who just share the same birth city since performing on the same work implies more similarities between these two artists. Therefore, the “collaboration” link based on the shared work in a LOD graph is likely to have a higher impact on relatedness than a “born in” link, and it should carry more weight for recommendation purposes. However, our baselines *LDSD* and *Resim* treat these cases equally and do not recognize the significance of some link paths that are more helpful for recommender systems. It is our belief that links should be distinguished based on the level of relatedness between resources indicated by the link. Therefore, we introduce weighted variations of our baselines (*WLDSD* and *WResim*) in this section to that prioritize different link types in LOD for recommender systems and then propose two different approaches to calculating the link weights, one based on probability theory (*RSLAW*) and the other based on information theory (*ITW*).

4.1.1 Weighted Approaches

4.1.1.1 Weighted Linked Data Semantic Distance (*WLDSD*)

A weighted version of the *LDSD* is introduced by including a weighting factor that modifies the semantic distance value based on link importance as an indicator of relatedness. This factor is introduced to the original *LDSD* defining the *Weighted Linked Data Semantic Distance (WLDSD)* as follows:

$$WLDSD_{cn}(r_a, r_b) = \frac{1}{1 + WDC'(r_a, r_b) + WDC'(r_b, r_a) + WIC'_i(r_a, r_b) + WIC'_o(r_a, r_b)}$$

where $WDC'(r_a, r_b)$ is the direct connectivity (*DC*) between resources r_a and r_b normalized by the log of all outgoing links from the resource r_a and weighted by the weight factor W_{l_j} for each link of type l_j as follows:

$$WDC'(r_a, r_b) = \sum_j \left(\frac{DLC(l_j, r_a, r_b)}{1 + \log(\sum_n DLC(l_j, r_a, r_n))} \times W_{l_j} \right)$$

$WDC'(r_b, r_a)$ is the direct connectivity (*DC*) between resources r_b and r_a normalized by the log of all outgoing links from the resource r_b and weighted by the weight factor W_{l_j} for each link of type l_j as follows:

$$WDC'(r_b, r_a) = \sum_j \left(\frac{DLC(l_j, r_b, r_a)}{1 + \log(\sum_n DLC(l_j, r_b, r_n))} \times W_{l_j} \right)$$

$WIC'_i(r_a, r_b)$ is the incoming indirect connectivity (IC_i) between resources r_a and r_b normalized by the log of all incoming indirect links to the resource r_a and weighted by the weight factor W_{l_j} for each link of type l_j as follows:

$$WIC'_i(r_a, r_b) = \sum_j \left(\frac{ILC_i(l_j, r_a, r_b)}{1 + \log(\sum_n ILC_i(l_j, r_a, r_n))} \times W_{l_j} \right)$$

$WIC'_o(r_a, r_b)$ is the outgoing indirect connectivity (IC_o) between resources r_a and r_b normalized by the log of all outgoing indirect links from the resource r_a and weighted by the weight factor W_{l_j} for each link of type l_j as follows:

$$WIC'_o(r_a, r_b) = \sum_j \left(\frac{ILC_o(l_j, r_a, r_b)}{1 + \log(\sum_n ILC_o(l_j, r_a, r_n))} \times W_{l_j} \right)$$

such that the value of every weight W_{l_j} is a positive rational number between zero and one ($0 \leq W_{l_j} \leq 1$).

The weighting factor W_{l_j} is introduced in the *LDSD* approach for every link-based operation. Therefore, higher direct and indirect connectivity values are generated for those links with a high weight (W_{l_j}); conversely, less emphasis is resulted on these links when their weight is low while some link types are cancelled if their corresponding weight is zero.

4.1.1.2 Weighted Resource Similarity (*WResim*)

The weighting factor is also introduced to the second baseline in this document, *Resim*, defining the *Weighted Resource Similarity (WResim)* as follows:

$$WResim(r_a, r_b) = \begin{cases} 0 & if\ URI(r_a) = URI(r_b)\ or\ r_a\ owl{:}\ sameAs\ r_b \\ WLDSD_\gamma(r_a, r_b) & if\ WLDSD_\gamma(r_a, r_b) \neq 1 \\ Property_{sim}(r_a, r_b) & otherwise \end{cases}$$

The *Weighted Linked Data Semantic Distance* ($WLDSD_\gamma$) component of *WResim* is calculated as follows:

$$WLDSD_\gamma(r_a, r_b) = \frac{1}{1 + WRC(r_a, r_b) + WRC(r_b, r_a) + WRI_i(r_a, r_b) + WRI_o(r_a, r_b)}$$

where $WRC(r_a, r_b)$ is simply the direct connectivity (*DC*) between resources r_a and r_b normalized by the log of number of instances of a link l_i weighted by the weighting factor W_{l_i} for each link of a type l_i as follows:

$$WRC(r_a, r_b) = \sum_i \left(\frac{DLC(l_i, r_a, r_b)}{1 + \log(\sum_m \sum_n DLC(l_i, r_m, r_n))} \times W_{l_i} \right)$$

$WRC(r_b, r_a)$ is the direct connectivity (*DC*) between resources r_b and r_a normalized by the log of number of instances of a link l_i weighted by the weighting factor W_{l_i} for each link of a type l_i as follows:

$$WRC(r_b, r_a) = \sum_i \left(\frac{DLC(l_i, r_b, r_a)}{1 + \log(\sum_m \sum_n DLC(l_i, r_m, r_n))} \times W_{l_i} \right)$$

$WRI_i(r_a, r_b)$ is the incoming indirect connectivity (IC_i) between resources r_a and r_b through a resource r_j normalized by the log of all incoming indirect links from the resource r_j with a link type of l_i weighted by the weighting factor W_{l_i} for each link of a type l_i as follows:

$$WRI_i(r_a, r_b) = \sum_i \sum_j \left(\frac{ILC_i(l_i, r_j, r_a, r_b)}{1 + \log(\sum_n ILC_i(l_i, r_j, r_n))} \times W_{l_i} \right)$$

$WRI_o(r_a, r_b)$ is the outgoing indirect connectivity (*IC*$_o$) between resources r_a and r_b through a resource r_j normalized by the log of all outgoing indirect links from the resource r_j with a link type of l_i weighted by the weighting factor W_{l_i} for each link of a type l_i as follows:

$$WRI_o(r_a, r_b) = \sum_i \sum_j \left(\frac{ILC_o(l_i, r_j, r_a, r_b)}{1 + \log(\sum_n ILC_o(l_i, r_j, r_n))} \times W_{l_i} \right)$$

such that the value of every weight W_{l_i} is a positive rational number between zero and one ($0 \leq W_{l_i} \leq 1$).

In addition, $Property_{sim}$ calculates the similarity of shared links types between resources r_a and r_b as previously mentioned in the original *Resim*. Similar to *WLDSD*, the weighting factor W_{l_i} is introduced in the *Resim* approach for every link-based operation and, therefore, higher direct and indirect connectivity values are generated for those links with a high weight (W_{l_i}).

4.1.2 Link Weights Calculation

The previous section raises a critical question: how to measure the weight of each link (W_{l_i})? This section introduces two approaches to this calculation: *Resource-Specific Link Awareness Weights* (*RSLAW*) and *Information Theoretic Weights* (*ITW*). The *RSLAW* weights are based on the association between each link type and its linked resources' classes whereas the *ITW* weights are based on the importance of the link to the resource along with its distribution in the LOD graph.

4.1.2.1 Resource-Specific Link Awareness Weights (*RSLAW*)

LOD resources are connected using different link types, and most of these types are used to connect different classes of resources. For example, the link type "*genre*" can be used to connect artists to their corresponding genre, and it can be used to link a song or a movie to its corresponding genre too. On the other hand, some link types tend to be very specific in connecting resources only within similar classes as in the case of the link type "*associatedMusicalArtist*" that is mostly used to connect musical artists to each other. It is our belief that link types that are typically used to link specific resource classes together indicate more relatedness than links used to connect a wide variety of resource classes. Based on this intuition, LOD link weights can be generated to emphasize those links that are specific to particular resources.

As *R* is already defined as the set of all resources in the linked data dataset, a recommender system can define a subset of *R* to indicate those resources that the recommender system is interested to include in the recommendation process. Formally, γ is a set of resources with a resource class intended for recommendation specified by the recommender system ($\gamma \subseteq R$).

In this approach, the weight of a link l_x is the probability that this link is associated with γ. In particular, the weight of a link l_x is the total number of instances of the link l_x between resources r_i and r_j that belong to a specific resource class set (γ) divided by the total number of instances of the link l_x between all resources regardless of their resource class as follows:

$$W_{l_x} = \frac{\sum_i \sum_j DLC(l_x, r_i, r_j)}{\sum_m \sum_n DLC(l_x, r_m, r_n)}, \quad \{\forall\, r_i, r_j \mid r_i \in \gamma \text{ and } r_j \in \gamma\}$$

To illustrate this approach, Table 1 shows the number of link type instances in a LOD dataset. The total number of instances of each link type in the whole dataset is shown in Table 1.a whereas Table 1.b shows the number of instances of each link type between specific resource classes only ("*dbpedia:MusicalArtist"* or "*dbpedia:MusicalBand*"), therefore γ is the set of all resources with class of ("*dbpedia:MusicalArtist"* or "*dbpedia:MusicalBand*"). There are three categories of link types in this example: highly resource-specific link types such as *associatedMusicalArtist* (95/100) and *associatedBand* (70/75), poorly resource-specific link types such as *influencedBy* (10/50) and *relative* (5/25), in addition to generic link types *occupation* (0/10), *hometown* (0/6). This approach prioritizes highly resource-specific link types as they carry more value between those resources whereas generic link types tend to describe general information about all classes of resources.

Table 1: RSLAW Example

a: Total link type instances

Link Type	Count
associatedMusicalArtist	100
associatedBand	75
influencedBy	50
relative	25
occupation	10
hometown	6

b: Resource-specific link type instances

Link Type	Count
associatedMusicalArtist	95
associatedBand	70
influencedBy	10
relative	5
occupation	0
hometown	0

4.1.2.2 Information Theoretic Weights (*ITW*)

The second approach to calculating the link weights is inspired from the well-known method from the information retrieval field, *TF-IDF* (Term Frequency–Inverse Document Frequency) **[49]**, which is used to weight the importance of a term in a document within a pool of documents. In our scope, the importance of a link to a resource is assessed based on the entire collection of resources and links in the LOD dataset. Unlike the *RSLAW* approach which considers only the links distribution in the dataset, the weights in this approach are dynamically calculated and take into consideration the relationship between the link and other links in the dataset in addition to the relationship between the link and the resources linked to. In contrast, the disadvantage of this approach is that the whole LOD dataset must be traversed in order to compute the weights for each link which can be heavy on computing resources. Yet, this value can be computed once and stored in a preprocessing step, then integrated into the LOD engine to use the weights on the fly as needed.

Since the weights are calculated dynamically, they are referred here as $\mathrm{W}(l_x, r_a, r_b)$ instead of W_{l_i} because they require all the link information for their calculation. Additionally, this approach results in weights that do not meet our proposed constraints of the link weights range ([0-1]); therefore, rescaling these values back into this range is required as discussed later in this section. Initially, the non-scaled information theoretic weights $\mathrm{W}_{ns}(l_x, r_a, r_b)$ are calculated as follows:

$$\mathrm{W}_{ns}(l_x, r_a, r_b) = LF(l_x, r_a, r_b) \times IRF(l_x, r_a, r_b)$$

In this formula[6], the link frequency $LF(l_x, r_a, r_b)$ is the average normalized frequency of a link l_x that connects either resources r_a or r_b to others. This normalized frequency is calculated as the total number of both incoming and outgoing links of a type l_x to either resource r_a or r_b normalized by the total number of both incoming and outgoing links to either resource r_a or r_b as follows:

$$LF(l_x, r_a, r_b) = \frac{\left(\frac{\sum_j DLC(l_x, r_a, r_j) + \sum_j DLC(l_x, r_j, r_a)}{\sum_i \sum_j DLC(l_i, r_a, r_j) + \sum_i \sum_j DLC(l_i, r_j, r_a)}\right) + \left(\frac{\sum_j DLC(l_x, r_b, r_j) + \sum_j DLC(l_x, r_j, r_b)}{\sum_i \sum_j DLC(l_i, r_b, r_j) + \sum_i \sum_j DLC(l_i, r_j, r_b)}\right)}{2}$$

The inverse resource frequency $IRF(l_x, r_a, r_b)$ is the total number of resources in the LOD dataset intended for recommendation ($|\gamma|$) divided by total instances of the link l_x as follows:

$$IRF(l_x, r_a, r_b) = \log \frac{\sum_i r_i}{\sum_i \sum_j DLC(l_x, r_i, r_j)}, \quad \{\forall\, r_i \mid r_i \in \gamma\}$$

Finally, weights calculated using this approach must be rescaled back in the range [0-1] as follows:

[6] We have tried different variations of this approach, and we report the best performing version only.

$$\mathrm{W}(l_x, r_a, r_b) = \frac{\mathrm{W}_{ns}(l_x, r_a, r_b) - min}{max - min}$$

where *min* is the value of the minimum calculated weight, and *max* is the maximum calculated weight value.

4.2 Typeless Semantic Distance

One of the advantages of the LOD is the massive amount of interconnected information, but the sheer volume of data causes several challenges. One of these challenges is that data accuracy in LOD can vary from one dataset to another and even within a given dataset. Several LOD datasets, including *DBpedia*, have their data collected and linked via human effort. For example, a link in *DBpedia* that represents the relationship between a song and its album can have different names (labels) such as "*fromAlbum*" or "*title*" depending on the editor who updated the song or album page in *Wikipedia*. However, when applying LOD in recommender systems, the recommender system must be able to recommend items even if the resources to be recommended are connected using different types of links. Therefore, it may be necessary for a recommendation to consider the relationship between resources even when their links have different types. This is especially true when mining relationship from multiple LOD datasets, each of which may have its own ontology or set of link types. Despite this, the indirect connectivity (*IC*) algorithm of our baselines does not consider these cases when calculating the indirect connectivity. In our work, we asses extending indirect connectivity calculations to include the effect of multiple links of differing types within Linked Open Data. This extension can be incorporated within both baselines to measure the effect of including heterogeneous link types in the semantic distance calculation.

Following the example in Figure 2, the outgoing indirect connectivity between r_1 and r_4 is zero ($IC_o(r_1, r_4) = 0$) since there is no resource such that both r_1 and r_4 are directly connected to through the same link type. However, both r_1 and r_4 are directly connected to r_2 but this is via different link types (l_1 and l_2). In our extension, we develop a typeless incoming and outgoing indirect connectivity between two resources r_a and r_b to broaden the indirect connectivity to include cases where the two resources can be connected by two different link types (l_k and l_p) as displayed in Figure 4. Formally, the *incoming typeless indirect connectivity*, *TIC$_i$*, between two resources r_a and r_b is the sum of the *incoming typeless indirect link connectivity*, *TILC$_i$*, of all links that connect them as follows:

$$TIC_i(r_a, r_b) = \sum_n \sum_j \sum_k TILC_i\left(l_j, l_k, r_n, r_a, r_b\right)$$

The *Incoming Typeless Indirect Link Connectivity* (*TILC$_i$*) between two resources r_a and r_b is equal to one if there is a resource r_n such that r_n is directly connected to both r_a and r_b via links of type l_k and l_p as follows:

$$TILC_i\left(l_k, l_p, r_n, r_a, r_b\right) = \begin{cases} 1 & \left\{\exists\ r_n \middle|\ \langle l_k, r_n, r_a\rangle \& \langle l_p, r_n, r_b\rangle\ \right\} \\ 0 & \text{otherwise} \end{cases}$$

Similarly, the *outgoing typeless indirect connectivity*, *TIC$_o$*, between two resources r_a and r_b is the sum of the *outgoing typeless indirect link connectivity*, *TILC$_o$*, of all links that connect them as follows:

$$TIC_o(r_a, r_b) = \sum_n \sum_j \sum_k TILC_o\left(l_j, l_k, r_n, r_a, r_b\right)$$

The *Outgoing Typeless Indirect Link Connectivity* (*TILC$_o$*) between two resources r_a and r_b is equal to one if there is a resource r_n such that both r_a and r_b are directly connected to r_n via links of type l_k and l_p as follows:

$$TILC_o\left(l_k, l_p, r_n, r_a, r_b\right) = \begin{cases} 1 & \left\{\exists\ \ r_n \middle|\ \langle l_k, r_a, r_n\rangle \& \langle l_p, r_b, r_n\rangle\ \right\} \\ 0 & \text{otherwise} \end{cases}$$

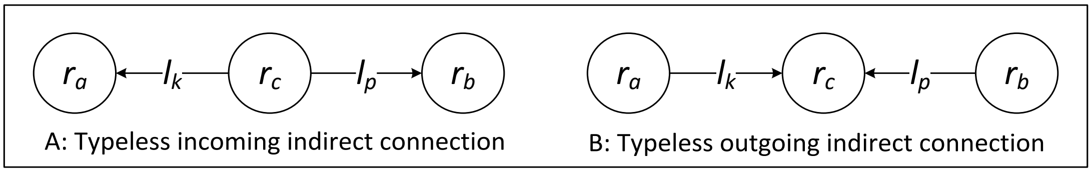

r_a
l_k
r_c
l_p
r_b
A: Typeless incoming indirect connection
r_a
l_k
r_c
l_p
r_b
B: Typeless outgoing indirect connection

Figure 4: Typeless indirect connectivity

Even though $IC_o(r_1, r_4) = 0$ as previously mentioned, the outgoing typeless indirect connectivity between r_1 and r_4 is one ($TIC_o(r_1, r_4) = 1$) through the resource r_2 with the links types (l_1, l_2), which shows that r_1 and r_4 are indirectly connected to each other with the typeless variation.

The typeless indirect link connectivity (*TILC*) notation can be generalized for all intermediate resources as follows[7]:

$$TILC_i(l_k, l_p, r_a, r_b) = \sum_n \sum_j \sum_k TILC_i(l_j, l_k, r_n, r_a, r_b)$$

$$TILC_o(l_k, l_p, r_a, r_b) = \sum_n \sum_j \sum_k TILC_o(l_j, l_k, r_n, r_a, r_b)$$

This concept can be applied to our baselines resulting in two typeless versions: *Typeless Linked Data Semantic Distance* (*TLDSD*) and *Typeless Resource Similarity (TResim)*.

[7] These versions of *TILC* accept four inputs instead of five as in the regular *TILC*

4.2.1 Typeless Linked Data Semantic Distance (*TLDSD*)

Based on the typeless indirect connectivity, a typeless version of *LDSD* is calculated as follows:

$$TLDSD_{cn}(r_a, r_b) = \frac{1}{1 + DC'(r_a, r_b) + DC'(r_b, r_a) + TIC'_i(r_a, r_b) + TIC'_o(r_a, r_b)}$$

where $DC'(r_a, r_b)$ is merely the direct connectivity (*DC*) between resources r_a and r_b normalized by the log of all outgoing links from the resource r_a as follows:

$$DC'(r_a, r_b) = \sum_j \frac{DLC(l_j, r_a, r_b)}{1 + \log(\sum_n DLC(l_j, r_a, r_n))}$$

$DC'(r_b, r_a)$ is the direct connectivity (*DC*) between resources r_b and r_a normalized by the log of all outgoing links from the resource r_b as follows:

$$DC'(r_b, r_a) = \sum_j \frac{DLC(l_j, r_b, r_a)}{1 + \log(\sum_n DLC(l_j, r_b, r_n))}$$

$TIC'_i(r_a, r_b)$ is the incoming typeless indirect connectivity (TIC_i) between resources r_a and r_b normalized by the log of all incoming typeless indirect links to the resource r_a as follows:

$$TIC'_i(r_a, r_b) = \sum_j \sum_k \frac{TILC_i(l_j, l_k, r_a, r_b)}{1 + \log(\sum_n TILC_i(l_j, l_k, r_a, r_n))}$$

$TIC'_o(r_a, r_b)$ is the outgoing typeless indirect connectivity (TIC_o) between resources r_a and r_b normalized by the log of all outgoing typeless indirect links from the resource r_a as follows:

$$TIC'_o(r_a, r_b) = \sum_j \sum_k \frac{TILC_o(l_j, l_k, r_a, r_b)}{1 + \log(\sum_n TILC_o(l_j, l_k, r_a, r_n))}$$

4.2.2 Typeless Resource Similarity (*TResim*)

Similar to *TLDSD*, a typeless version of *Resim* is defined as follows:

$$TResim(r_a, r_b) = \begin{cases} 0 & if\ URI(r_a) = URI(r_b)\ or\ r_a\ owl\!:\!sameAs\ r_b \\ TLDSD_\gamma(r_a, r_b) & if\ TLDSD_\gamma(r_a, r_b) \neq 1 \\ Property_{sim}(r_a, r_b) & otherwise \end{cases}$$

The *Typeless Linked Data Semantic Distance* ($TLDSD_\gamma$) component is calculated as follows:

$$TLDSD_\gamma(r_a, r_b) = \frac{1}{1 + RC(r_a, r_b) + RC(r_b, r_a) + TRI_i(r_a, r_b) + TRI_o(r_a, r_b)}$$

where $RC(r_a, r_b)$ is the direct connectivity (*DC*) between resources r_a and r_b normalized by the log of number of instances of a link l_j as follows:

$$RC(r_a, r_b) = \sum_j \frac{DLC(l_j, r_a, r_b)}{1 + log(\sum_m \sum_n DLC(l_j, r_m, r_n))}$$

$RC(r_b, r_a)$ is the direct connectivity (*DC*) between resources r_b and r_a normalized by the log of number of instances of a link l_j as follows:

$$RC(r_b, r_a) = \sum_j \frac{DLC(l_j, r_b, r_a)}{1 + log(\sum_m \sum_n DLC(l_j, r_m, r_n))}$$

$TRI_i(r_a, r_b)$ is the incoming typeless indirect connectivity (*TIC_i*) between resources r_a and r_b through a resource r_k normalized by the log of all incoming typeless indirect links from the resource r_k as follows:

$$TRI_i(r_a, r_b) = \sum_k \sum_i \sum_j \left(\frac{TILC_i(l_i, l_j, r_k, r_a, r_b)}{1 + \log(\sum_n TILC_i(l_i, l_j, r_k, r_n))} \right)$$

$TRI_o(r_a, r_b)$ is the outgoing typeless indirect connectivity (*TIC_o*) between resources r_a and r_b through a resource r_k normalized by the log of all outgoing typeless indirect links from the resource r_k as follows:

$$TRI_o(r_a, r_b) = \sum_k \sum_i \sum_j \left(\frac{TILC_o(l_i, l_j, r_k, r_a, r_b)}{1 + \log(\sum_n TILC_o(l_i, l_j, r_k, r_n))} \right)$$

In addition, $Property_{sim}$ calculates the similarity of shared links types between resources r_a and r_b as previously mentioned in the original *Resim*.

4.3 Weighted Typeless Semantic Distance

After we defined weighted variations of our baselines as well as the typeless variations, we investigate combining the approaches of weighting links and typeless links to evaluate the effects of prioritizing some link paths in LOD regardless of their type in the indirect connectivity. This combined approach results in two new variations of the baselines: *Weighted Typeless Linked Data Semantic Distance (WTLDSD)* and *Weighted Typeless Resource Similarity (WTResim)*.

4.3.1 Weighted Typeless Linked Data Semantic Distance (*WTLDSD*)

The weighted approach can be applied to the typeless version of *LDSD* as follows:

$$WTLDSD_{cn}(r_a, r_b) = \frac{1}{1 + WDC'(r_a, r_b) + WDC'(r_b, r_a) + WTIC'_i(r_a, r_b) + WTIC'_o(r_a, r_b)}$$

where $WDC'(r_a, r_b)$ is the direct connectivity (*DC*) between resources r_a and r_b normalized by the log of all outgoing links from the resource r_a and weighted by the weight factor W_{l_j} for each link of a type l_j as follows:

$$WDC'(r_a, r_b) = \sum_j \left(\frac{DLC(l_j, r_a, r_b)}{1 + \log(\sum_n DLC(l_j, r_a, r_n))} \times W_{l_j} \right)$$

$WDC'(r_b, r_a)$ is the direct connectivity (*DC*) between resources r_b and r_a normalized by the log of all outgoing links from the resource r_b and weighted by the weight factor W_{l_j} for each link of a type l_j as follows:

$$WDC'(r_b, r_a) = \sum_j \left(\frac{DLC(l_j, r_b, r_a)}{1 + \log(\sum_n DLC(l_j, r_b, r_n))} \times W_{l_j} \right)$$

$WTIC'_i(r_a, r_b)$ is the incoming typeless indirect connectivity (TIC_i) between resources r_a and r_b normalized by the log of all incoming typeless indirect links to the resource r_a and weighted by W_{l_j} or W_{l_k} for each link of types l_j or l_k correspondingly as follows:

$$WTIC'_i(r_a, r_b) = \sum\nolimits_j \left(\sum\nolimits_k \frac{TILC_i(l_j, l_k, r_a, r_b)}{1 + \log(TILC_i(l_j, l_k, r_a, n_r))} \times W_{l_k} \right) \times W_{l_j}$$

$WTIC'_o(r_a, r_b)$ is the outgoing typeless indirect connectivity (*TIC_o*) between resources r_a and r_b normalized by the log of all outgoing typeless indirect links from the resource r_a and weighted by W_{l_j} or W_{l_k} for each link of types l_j or l_k correspondingly as follows:

$$WTIC'_o(r_a, r_b) = \sum\nolimits_j \left(\sum\nolimits_k \frac{TILC_i(l_j, l_k, r_a, r_b)}{1 + \log(TILC_i(l_j, l_k, r_a, n_r))} \times W_{l_k} \right) \times W_{l_j}$$

The value of every weight W_{l_j} or W_{l_k} is a positive rational number between zero and one $(0 \leq W_{l_j} \leq 1)$ & $(0 \leq W_{l_k} \leq 1)$.

4.3.2 Weighted Typeless Resource Similarity (*WTResim*)

Similar to *WTLDSD*, a typeless version of *WResim* is defined as follows:

$$WTResim(r_a, r_b) = \begin{cases} 0 & if\ URI(r_a) = URI(r_b)\ or\ r_a\ owl{:}sameAs\ r_b \\ WTLDSD_\gamma(r_a, r_b) & if\ WTLDSD_\gamma(r_a, r_b) \neq 1 \\ Property_{sim}(r_a, r_b) & otherwise \end{cases}$$

The weighted typeless linked data semantic distance ($WTLDSD_\gamma$) component is calculated as follows:

$$WTLDSD_\gamma(r_a, r_b) = \frac{1}{1 + WRC(r_a, r_b) + WRC(r_b, r_a) + WTRI_i(r_a, r_b) + WTRI_o(r_a, r_b)}$$

where $WRC(r_a, r_b)$ is simply the direct connectivity (*DC*) between resources r_a and r_b normalized by the log of number of instances of a link l_i weighted by the weighting factor W_{l_i} for each link of a type l_i as follows:

$$WRC(r_a, r_b) = \sum_i \left(\frac{DLC(l_i, r_a, r_b)}{1 + \log(\sum_m \sum_n DLC(l_i, r_m, r_n))} \times W_{l_i} \right)$$

$WRC(r_b, r_a)$ is the direct connectivity (*DC*) between resources r_b and r_a normalized by the log of number of instances of a link l_i weighted by the weighting factor W_{l_i} for each link of a type l_i as follows:

$$WRC(r_b, r_a) = \sum_i \left(\frac{DLC(l_i, r_b, r_a)}{1 + \log(\sum_m \sum_n DLC(l_i, r_m, r_n))} \times W_{l_i} \right)$$

$WTRI_i(r_a, r_b)$ is the incoming typeless indirect connectivity (TIC_i) between resources r_a and r_b through a resource r_k normalized by the log of all incoming typeless indirect links from the resource r_k , and weighted by W_{l_i} or W_{l_j} for each link of types l_j or l_k correspondingly as follows:

$$WTRI_i(r_a, r_b) = \sum_k \left(\sum_i \left(\sum_j \left(\frac{TILC_i(l_i, l_j, r_k, r_a, r_b)}{1 + \log(\sum_n TILC_i(l_i, l_j, r_k, r_n))} \right) \times W_{l_j} \right) \times W_{l_i} \right)$$

$WTRI_o(r_a, r_b)$ is the outgoing typeless indirect connectivity (TIC_o) between resources r_a and r_b through a resource r_k normalized by the log of all outgoing typeless indirect links from the resource r_k , and weighted by W_{l_i} and W_{l_j} for each link of types l_j or l_k correspondingly as follows:

$$WTRI_o(r_a, r_b) = \sum_k \left(\sum_i \left(\sum_j \left(\frac{TILC_o(l_i, l_j, r_k, r_a, r_b)}{1 + \log(\sum_n TILC_o(l_i, l_j, r_k, r_n))} \right) \times W_{l_j} \right) \times W_{l_i} \right)$$

The value of every weight W_{l_i} or W_{l_j} is a positive rational number between zero and one $(0 \le W_{l_i} \le 1)$ & $(0 \le W_{l_j} \le 1)$. In addition, $Property_{sim}$ calculates the similarity of shared links types between resources r_a and r_b as previously mentioned in the original *Resim*.

5 Employing Semantic Distance Approaches for Multiple Nodes Apart Resources in LOD

Linked Open Data is rich with resources related to each other but they are not always directly linked or indirectly linked via a single hub. Resources that are further apart within the network could be important to recommend. However, one drawback of the *LDSD* approach is that it only calculates the semantic distance between resources that are either directly connected or indirectly connected through an intermediate resource. Therefore, resources that are located more two links away are automatically considered unrelated to each other. *Resim* improves upon this calculating a simpler semantic relatedness between resources more than two links away based upon their properties. However, this calculation ignores the graph structure for these more distance resources altogether. For example, Figure 5 illustrates an example of a snapshot of a LOD dataset with five resources. In this example, resources r_2 and r_3 are reachable to the resource r_1 in *LDSD*, however, resources r_4 and r_5 are not reachable to the resource r_1 and therefore are considered unrelated to r_1.

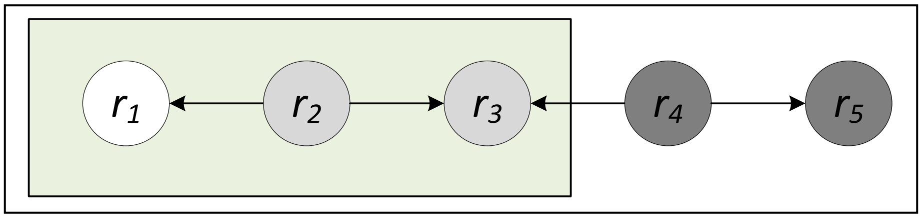


Figure 5: Example of reachable resources in *LDSD*

In this document, we introduce an approach that expands the coverage of current semantic distance approaches to resources that are linked through more than one intermediate hub. This approach is beneficial in several ways. First, we are able to create a much fuller collection of related resources for isolated resources that have sparse connections to others. In particular, LOD-based recommender systems performance has a strong correlation with the number of resource links as their accuracy declines for sparse resources [10]. Thus, propagating semantic connections further through the network of LOD expands resource coverage and may lead to a higher recall. Additionally, even for well-connected resources, propagating connections more widely may allow us to recommend related resources from another domain, e.g., link from a book to a related movie.

To achieve this goal, we employ an all-pairs shortest path algorithm, namely, the well-known Floyd-Warshall algorithm [50] to propagate semantic connectivity weights throughout the network of connected resources. This algorithm may not just increase the span of the semantic distance calculations; it also may increase the accuracy of the semantic distance calculations. This section is under publishing at [51].

5.1 Design

Incorporating more than one intermediate node in semantic distance calculations is challenging, especially with respect to efficiency. As we propagate weights throughout the network, the time complexity undergoes combinatorial explosion. Computing all semantic connection weights has an upper bound of $O(n^n)$ where n is the number of resources in the network, clearly intractable in LOD since it contains millions of nodes. The principle of our approach is to calculate the semantic distance between each linked resource pair in γ, and then propagate these values using an all-pair shortest path algorithm to get the final semantic distance values between all pairs. This approach first reduces the original graph to include only those resources that are under consideration by the recommender system, and then it calculates final semantic distances using this reduced graph. Figure 6 illustrates this proposed propagated approach based on the Floyd–Warshall algorithm for calculating an all-pair shortest path in graphs. The time complexity of this algorithm for both average and worst-case performance is $\Theta(| \gamma |^3)$.

The first step in our approach is to create an | γ | × | γ | matrix (assuming that resources are labeled from 1 to | γ |). Since semantic distances range from 0.0 to 1.0, this matrix is initialized with either 0.0 for the distance from each resource to itself, 1.0 otherwise. Unlike the original Floyd-Warshall algorithm in which the matrix is initialized with infinity (∞), the value 1.0 here is the maximum semantic distance referring value that reflects the lack of any relatedness.

```
1  let d be a $|\gamma| \times |\gamma|$ array of minimum semantic distances
2  for i from 1 to $|\gamma|$
3   for j from 1 to $|\gamma|$
4    if i==j
5     d[i][j]=0
6    else
7     d[i][j]=1
8  for each resource pair ($r_a$, $r_b$) $\in |\gamma|$
9   d[$r_a$][$r_b$] = SemanticDistance($r_a$, $r_b$)
10 for k from 1 to $|\gamma|$
11  for i from 1 to $|\gamma|$
12   for j from 1 to $|\gamma|$
13    if  d[i][j] > 1- ((1- d[i][k]) × (1- d[k][j]))
14     d[i][j] = 1- ((1- d[i][k]) × (1- d[k][j]))
15    end if
```

Figure 6: The propagated semantic distance algorithm

5.2 LOD Graph Reduction

After initializing the semantic distance matrix (d), the semantic distance is calculated between each resource pair that is an element of γ (lines 8 and 9 in Figure 6). This semantic distance calculation can use any semantic distance approach including *LDSD* or *Resim*. This step results in a reduced graph consisting only of γ resources instead of the whole LOD graph. For example, Figure 7 shows an example of a snapshot of a LOD dataset. In this example, resources r_1, r_3 and r_5 have the same resource class (e.g. *MusicalArtist*) that is a subset of γ while resource r_2 and r_4 have different resource classes (e.g. *Album* or *MusicalWork*) that are not subsets of γ. Therefore, this approach calculates the semantic distance between resources r_1, r_3 and r_5 only and results in semantic distance values between these pairs. Yet, resources r_2 and r_4 contribute to the semantic distance calculation because resources r_1, r_3 and r_5 are indirectly linked through these resources.

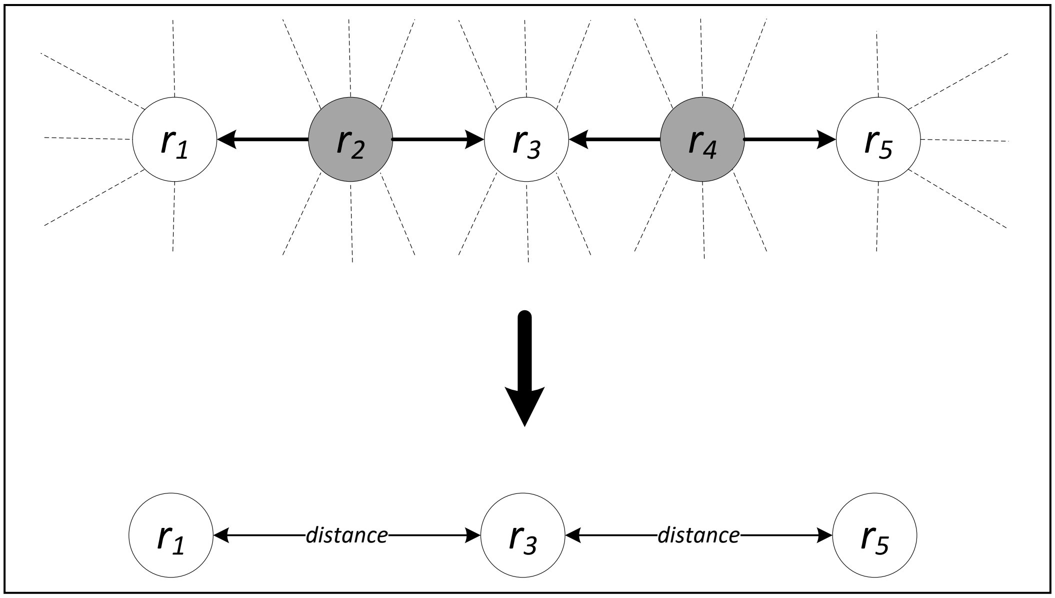


Figure 7: LOD graph reduction example

5.3 Semantic Distance Propagation

After obtaining semantic distance values between all resources pairs as shown in lines 8 and 9 of the algorithm, the Floyd–Warshall algorithm is applied to compare all possible paths in the reduced graph to find the optimal path that achieves the lowest semantic distance value. The intuition behind this propagation is that relatedness can be propagated through resources taking into account that semantic distance values reflect this propagation. For instance, if a resource r_a is 50% related to a resource r_b, and the resource r_b is 50% related to a resource r_c then the resource r_a is 25% related to the resource r_c (50% of the 50%). The Floyd–Warshall algorithm is feasible in this application since semantic distance values are positive values ranging from zero to one where zero represents 100% relatedness while the value one represents no relatedness at all.

The semantic distance matrix is updated by considering all resources as an intermediate resource. This algorithm considers all resources one by one and updates all shortest paths including the current resource as an intermediate resource. In the algorithm, there are three phases:

1. the intermediate resource (*k*) iteration as in line 10
2. the source resource (*i*) iteration as in line 11
3. the destination resource (*j*) iteration as in line 12

When an intermediate resource (*k*) is picked between resources *i* and *j*, it can contribute to a lower semantic distance if the semantic distance value through it is lower than the current one (lines 13 and 14). Unlike the original Floyd-Warshall that deals with distances as integer numbers, our comparison takes into consideration semantic distance values that range from 0.0 to 1.0. Also, semantic distances propagation is performed via multiplication so that semantic connectivity loss is proportional to the amount of propagation in the network from the original resources.

6 System Architecture

The proposed approaches are implemented and evaluated in a real world scenario in order to measure their effectiveness. A content-based musical recommender system is implemented to recommend musical artists or bands based on the popular LOD dataset, *DBpedia*. The recommendation results of this recommender system are based on the semantic distance between resources (musical artists or bands) calculated using the various proposed semantic distance approaches. The following section details the components of this system.

6.1 System Components

We designed and implemented the recommender system to work with any resource class in the LOD ontology, thus it is capable of being applied in any domain. However, we tailored the current version of the recommender system to recommend musical artists or bands as a proof of concept. Figure 8 diagrams the system components. Each component is described in more detail in the following sections.

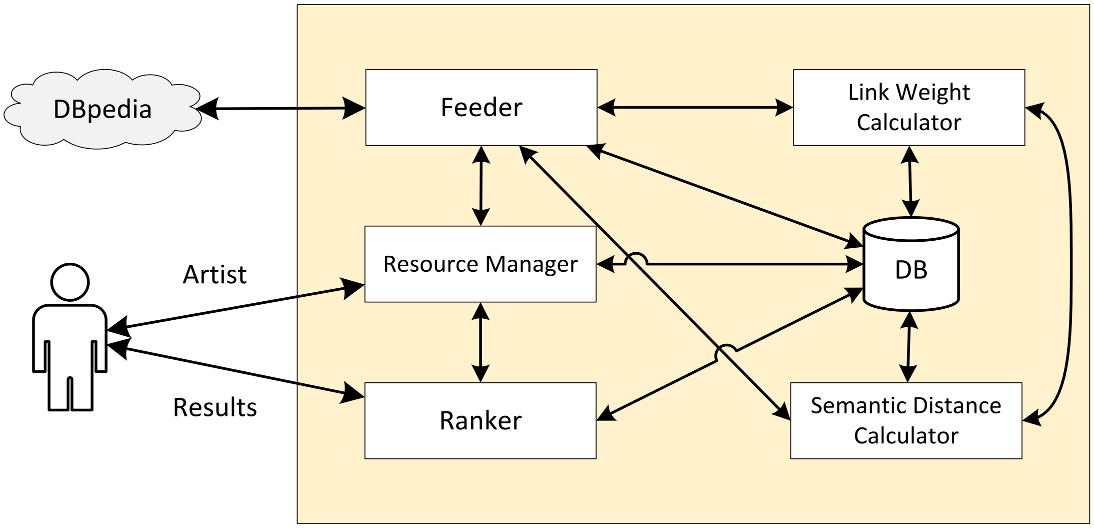


Figure 8: System architecture

6.1.1 Feeder

The *Feeder* component retrieves all the required data used by all other components from *LOD* providers such as *DBpedia*. This component requests all the desired data using the query language, *SPARQL Protocol and RDF Query Language (acronymed as SPARQL)*. The Feeder component receives the requested data from the LOD provider in *Resource Description Framework (RDF)* format and then converts it to our system's appropriate internal data structures. The Feeder component also caches some data request responses in the system's database in order to increase the efficiency of data gathering by avoiding duplicate requests.

6.1.2 Link Weight Calculator

The *Link Weight Calculator* component is responsible for computing the weight of every link type in the LOD dataset so that these weights can be used for the various semantic distance calculations between resources. This component runs independently of the interactive recommender system and it performs its calculations before any recommendations can be produced. It traverses the LOD dataset through the *Feeder* component and calculates the link weights based on the weighting approach being employed. Finally, it stores the weights of each weighting approach for every link in the system database to be used later by the *Semantic Distance Calculator* component.

6.1.3 Semantic Distance Calculator

The *Semantic Distance Calculator* component calculates the semantic distance between all the resources in order to be used later by the *Ranker* component. This component runs after the *Link Weight Calculator* component completes its work. This component traverses the LOD dataset through the *Feeder* component and stores the semantic distances calculated by the various approaches being employed in the system database.

6.1.4 Resource Manager

The *Resource Manager* component is responsible for searching the desired resources (musical artists or bands) in the LOD provider via the *Feeder* component. It retrieves their *Uniform Resource Identifiers (URI)* that uniquely identify resources in the system. It also ensures that semantic distances between the desired resource and all the other resources are already calculated by the *Semantic Distance Calculator* component and stored in the system database. Then, it forwards the *URI* to the *Ranker* component.

6.1.5 Ranker

The *Ranker* component retrieves and ranks related resources based on the semantic distance already calculated by the *Semantic Distance Calculator* component. The *Ranker* component generates a list of recommended resources for each user based on the similarity between the user's profile and every resource in the dataset. The similarity score between user u_i and resource *ra* is calculated based on the semantic distance generated by the various approaches we are evaluated as in the following:

$$\text{similarity}(u_i, r_a) = \frac{\sum_{r_b \in Profile(u_i)} (1 - SemanticDistance(r_a, r_b))}{|Profile(u_i)|}$$

where $Profile(u_i)$ is the user profile containing a list of resources that a user u_i has liked. $SemanticDistance(r_a, r_b)$ is the semantic distance between resources r_a and r_b based on the various semantic distance approaches.

The resulting list of resources is then sorted in a descending order and presented to the user.

6.2 System Environment

The recommender system is implemented in *Java* programming language. Both the *Weight Calculator* and *Distance Calculator* components are implemented as individual *Java* applications for efficiency reasons since they can run independently of the other system components. The database of the system is hosted in the database engine, *MySQL*. The entire system is cross-platform as *Java* and *MySQL* work on all popular operating systems such as Microsoft Windows and Linux.

7 Evaluation

In order to assess whether or not our proposed approaches are effective, we conducted several experiments to measure their effectiveness against three baselines *LDSD*, *Resim*, and *Jaccard Index*. The *Jaccard Index* [52], also called the Jaccard similarity coefficient, is a statistical measure to estimate the similarity between two sets. It is calculated as the number of items shared by the sets divided by the number of items in either set as follows:

$$\text{Jaccard}(r_a, r_b) = \frac{|N(r_a) \cap N(r_b)|}{|N(r_a) \cup N(r_b)|}$$

such that $N(r_a)$ is the set of neighbor resources to a resource $r_{a.}$ which is directly linked to each member of the set.

Similar to several related works in this field [9] [30] [33], we applied these experiments in the music domain to measure the relatedness between musical artists and bands. The following sections detail the dataset of the experiments followed by their methodology.

7.1 Dataset

We conducted the experiments using a dataset from the second Linked Open Data-enabled recommender systems challenge[8]. This dataset was built from Facebook profiles by collecting personal preferences (likes) for items (resources) in several domains. It contains the preferences of 52,069 users and includes 5,751 distinct resources in the music domain mapped to their corresponding resources in *DBpedia* (those with resources type "*dbpedia:MusicalArtist*" or "*dbpedia:MusicalBand*"). The total number of users' musical preferences was 1,013,973 with an average of 19.47 likes per user and a maximum of 37 likes per user. In addition, we calculated

[8] http://sisinflab.poliba.it/events/lod-recsys-challenge-2015/dataset/

the semantic distance between all resources in the previously mentioned dataset for all approaches (ours and baselines) on a live *DBpedia* server (version 2015-10)[9].

7.1.1 DBpedia

DBpedia [4] is a cross-domain LOD dataset with 400 million properties about more than 5 million resources ("things") extracted from *Wikipedia* pages. These resources are categorized into classes that construct the *DBpedia* ontology which contains around 750 classes [10]. Figure 9 displays a snapshot of the *DBpedia* ontology.

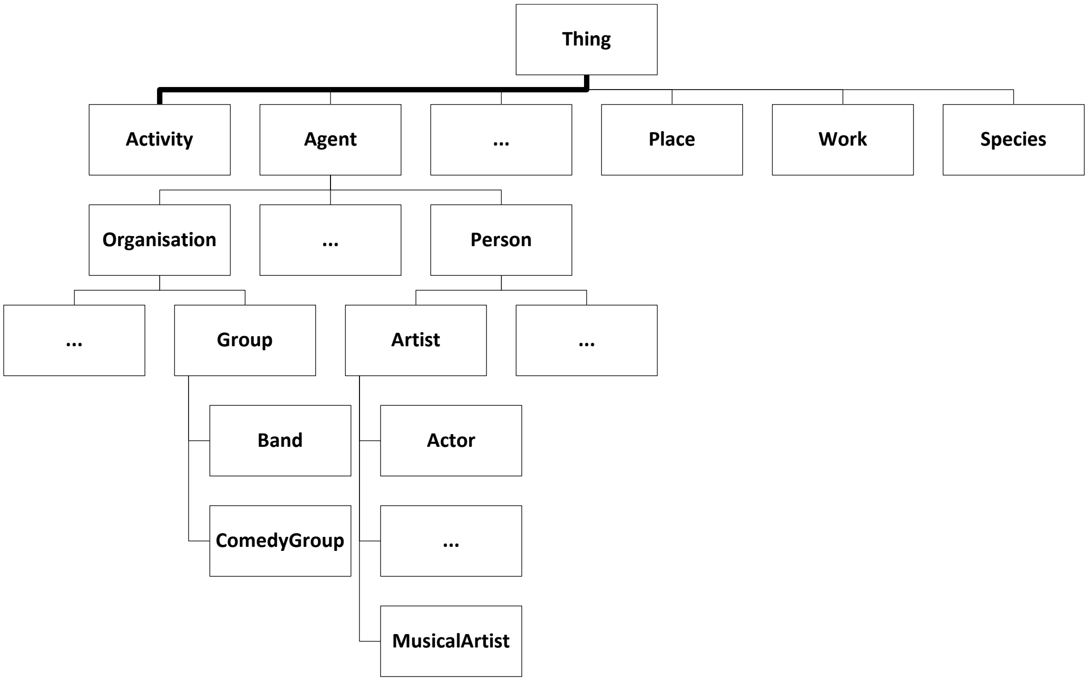


Figure 9: A snapshot of the *DBpedia* ontology

Table 2 shows the number of resources for some resource classes in *DBpedia*. There are around 1.5 million resources classified as *Person* in *DBpedia*. Fragment of these resources are

[9] http://wiki.dbpedia.org/Downloads2015-10
[10] http://wiki.dbpedia.org/services-resources/ontology

50,978 resources classified as musical artists (*dbo:MusicalArtist*) and 33,613 classified as musical bands (*dbo:Band*) making a total of 84,591 resources.

Table 2: *DBpedia* resources statistics

Resource Class	**# of resources**
Place	816,252
Person	1,517,816
Work	492,729
Species	301,025
Organisation	275,077
Other	1,706,991
Total	**5,109,890**

In *DBpedia*, there are 43 unique outgoing links types and 131 unique incoming links types from the *MusicalArtist* and *Band* classes. Table 3 shows a detailed link analysis for both resource classes in *DBpedia*. There are 39 unique link types outgoing from the resources class *MusicalArtist*, and 116 unique link types incoming to it. The average number of unique outgoing link types per resource with a *MusicalArtist* class is 3.91 while the average number of unique incoming link types per resource is 3.05. Also, the average number of the total outgoing links types per resource with a *MusicalArtist* class is 8.74 while it is 11.43 for incoming links. Similarly, there are 20 unique link types outgoing from the resources class *Band*, and 91 unique link types incoming to it. Furthermore, the average number of unique outgoing link types per resource with a *Band* class is 3.84 whereas the average number of unique incoming link types per resource is 3.08. The average of the total outgoing links types per resource with a *Band* class is 8.76 while it is 11.18 for incoming links.

Table 3: Link analysis of musical artists and bands in *DBpedia*

	dbo:MusicalArtist	*dbo:Band*	Both *
# of unique **outgoing** links types	39	20	43
Average # of unique **outgoing** links types per resource	3.91	3.84	3.88
Average # of total **outgoing** links types per resource	8.74	8.76	8.75
# of unique **incoming** links types	116	91	131
Average # of unique **incoming** links types per resource	3.05	3.08	6.64
Average # of total **incoming** links types per resource	11.43	11.18	11.33

* `{?resource a dbo:MusicalArtist } UNION {? resource a dbo:Band}`

7.2 Methodology

There are two ways to measure the recommendation accuracy of recommender systems: rating prediction and ranking [53]. The rating prediction method compares the prediction rating of a particular algorithm to ground truth, and it is often measured using the *Round Mean Square Error (RMSE)* or the *Mean Absolute Error (MAE)*. On the other hand, the ranking approach, also called top-k recommendation, compares a ranked list of recommended items to set aside items in a user profile using metrics such as precision, recall, F_1 score and *Mean Reciprocal Rank (MRR)*. Since we adopt the latter approach in this document, we describe each of the previously mentioned four metrics in the following subsections.

Similar to the approach taken by previous studies [10] [11], we randomly selected 500 users who have at least 10 preferences from the aforementioned dataset. Five preferences per user were reserved for testing purposes while the rest of their preferences, from a minimum of 5 to a maximum of 35 with an average of 19.22, were used to build a profile for each user u_i. The Profile is simply the set of resources liked by the user (represented by the resource ids) as follows:

$$Profile(u_i) = \{r_1, r_2, \ldots, r_m\}$$

where *m* is the number of resources in the training set that a user u_i has liked.

Next, we generated a list of recommended resources for each user based on the similarity between the user's profile and every resource in the dataset. The similarity score between user u_i and resource r_a is calculated based on the semantic distance generated by the various approaches we evaluated as in the following:

$$\text{similarity}(u_i, r_a) = \frac{\sum_{r_b \in Profile(u_i)} (1 - SemanticDistance(r_a, r_b))}{|Profile(u_i)|}$$

This essentially computes the probability that a user u_i appreciates the resource r_a by calculating the average semantic distance between each resource in the user profile and the specified resource.

This resulting list of resources was sorted in a descending order per user, and rank ordered list of recommended resources were compared to the ground truth by seeing where the user's liked resources in the testing dataset appeared within the list of recommendations. We measured the effectiveness of each semantic distance approach using the standard metrics of the F_1 Score and the *Mean Reciprocal Rank (MRR).*

7.2.1 F_1 Score

The F_1 score (also referred to as F-score or F-measure) [54] is a measurement of test accuracy that combines both precision and recall measurements. Precision, the percentage of good results in a set, is calculated as the number of correct positive results divided by the total number of all positive results.

$$precision = \frac{|true\ positive\ results|}{|true\ positive\ results| + |false\ positive\ results|}$$

Recall, the percentage of all good results that are presented to the users within the result set, is calculated as the number of correct positive results divided by the total number of actual positive results in the dataset.

$$recall = \frac{|true\ porsitive\ results|}{|true\ positive\ results|\ +\ |false\ negative\ results|}$$

The F_1 score, the harmonic mean of precision and recall, is calculated as follows:

$$F_1 = 2\times\frac{precision\ \times recall}{precision + recall}$$

7.2.2 Mean Reciprocal Rank (MRR)

Precision, recall, and F_1 all treat the results as a set, ignoring the order of presentation of the good results within a rank ordered list. Thus, many researchers prefer to use the *Mean Reciprocal Rank (MRR)* that takes into account how early a relevant result appears within ranked results. MRR is calculated as follows [55]:

$$MRR = \frac{\sum_{i=1}^{|Q|}\frac{1}{rank_i}}{|Q|}$$

where $rank_i$ is the highest rank of relevant results in a query Q_i.

7.3 Experiment 1: Effects of Weighting Links in Semantic Distance

In this experiment, we evaluate the effect of differentially weighting link types on recommendation accuracy. We calculated the link weights using three techniques:

1) Probability-based weights for each link type (*RSLAW*)
2) Information theory-based weights for each link type (*ITW*)
3) Random weights

We compared each of the three link-weighting schemes to three baselines, *LDSD* and *Resim*, and *Jaccard Index*, all of which weight all links identically. Table 4 shows the results of the experiment evaluated using the F_1 score and MRR metrics. The F_1 score values are presented at different ranked results cutoffs, i.e., 5, 10, and 20. In this table, the best results are shown in bold.

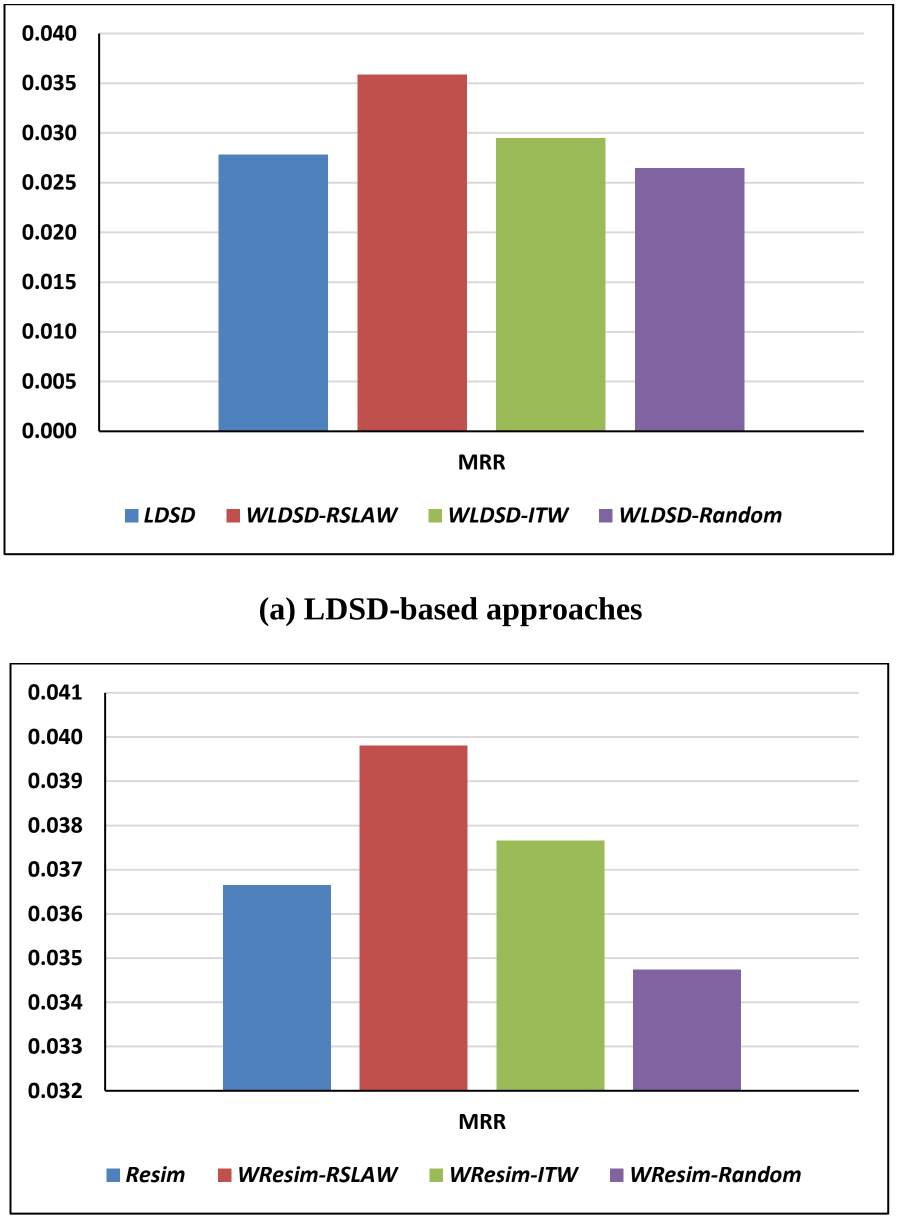
Table 4: Experiment 1 results for weighted approaches vs baselines

		LDSD-based Approaches				Resim-based Approaches			
	Jaccard	*LDSD*	Weighted (*WLDSD*)			*Resim*	Weighted (*WResim*)		
			RSLAW	*ITW*	*Random*		*RSLAW*	*ITW*	*Random*
MRR	0.010	0.028	0.036	0.029	0.026	0.037	**0.040**	0.038	0.035
F_1@5	0.009	0.031	0.041	0.033	0.016	0.049	**0.052**	0.049	0.046
F_1@10	0.011	0.044	0.048	0.045	0.043	0.053	**0.054**	**0.054**	0.048
F_1@20	0.012	0.046	0.053	0.052	0.051	0.051	**0.054**	0.051	0.046

The first conclusion we can draw from this experiment is that our basic baseline, *Jaccard Index*, indeed scored the lowest among all the metrics (F_1 and MRR). The MRR of the *Jaccard Index* was 0.010 whereas it was 0.028 and 0.037 for *LDSD* and *Resim* respectively. We can confirm that *LDSD* and *Resim* approaches which take resource connections in the LOD graph into considerations performs better in LOD resource similarity approaches.

Second, the *Resim* baseline outperforms the *LDSD* baseline among all metrics (MRR of 0.037 for *Resim* versus 0.028 for *LDSD*), confirming results reported by [10].

Third, our weighted approaches using either *RSLAW* or *ITW* weights outperform all baselines *Jaccard Index*, *LDSD* and *Resim* in all metrics. These improvements were statistically significant ($p<0.05$) based on a paired student t-test. As seen in Figure 10, the MRR was 0.036 for our *WLDSD-RSLAW* approach versus 0.028 for the non-weighted *LDSD* approach, an improvement of 29%, whereas it was 0.040 for *WResim-RSLAW* versus 0.037 for the original *Resim*, an 8% improvement.

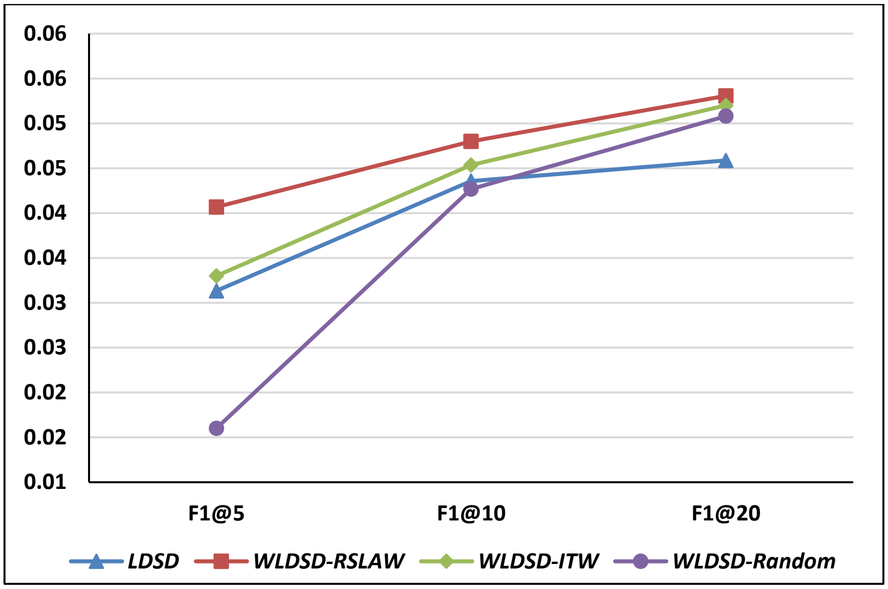


(a) LDSD-based approaches

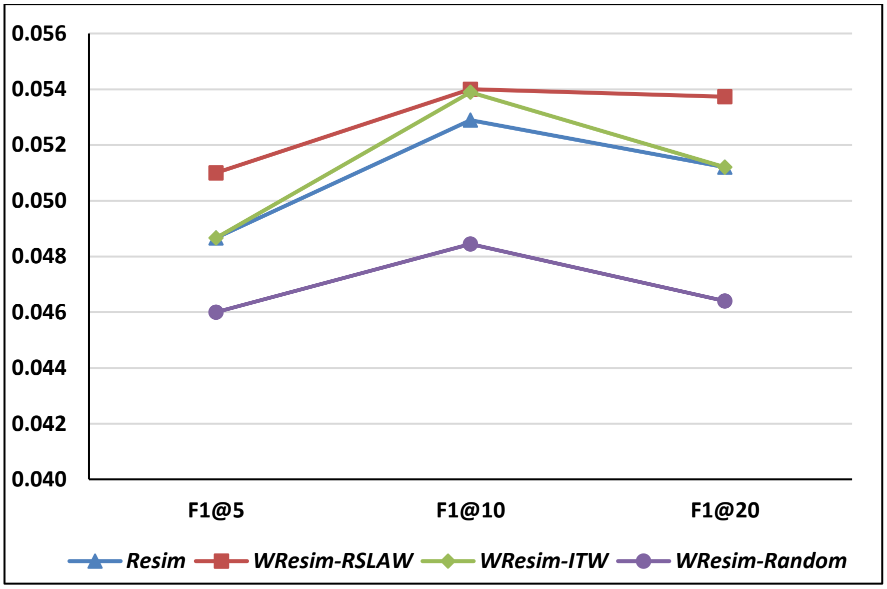
(b) Resim-based approaches

Figure 10: MRR scores for weighted approaches vs baselines

Confirming the MRR metric, the F_1 score of *WLDSD-RSLAW* was 0.041 for the top five results versus a score of 0.031 for *LDSD*, an improvement of 32% while it was 0.052 for *WResim-RSLAW* versus 0.049 versus the unweighted *Resim*, an improvement of 6%. These results also hold at other results cutoff points as displayed in Figure 11. Even though the improvement rate in *WResim-RSLAW* is not as large as it is in *WLDSD-RSLAW*, it achieved the most accurate recommendation results among all approaches in the experiment.



(a) LDSD-based approaches



(b) Resim-based approaches

Figure 11: F_1 scores at different ranked results cutoffs for weighted approaches vs baselines

The fourth conclusion we can draw from this experiment is that the *RSLAW* weights produced a bigger improvement than the *ITW* weights in both weighted approaches *WLDSD* and *WResim*. The information theoretic weights approach (ITW) performed slightly better than the baselines, *LDSD* (MRR of 0.029 vs 0.028) and *Resim* (MRR of 0.038 vs 0.037) but worse than the *RSLAW* weighting approach in general. Even though it was not as accurate as the *RSLAW* approach, it still confirms the importance of exploiting the link types in order to achieve better recommendation results.

Lastly, the experiment results also demonstrate that using random weights in both *WLDSD-Random* and *WResim-Random* results in reduced accuracy against both baselines *LDSD* (MRR of 0.026 vs 0.028) and *Resim* (MRR of 0.035 vs 0.037). This observation also holds at all the F_1 results cutoffs (@5, @10, and @20), and it confirms that the higher accuracy achieved by the *RSLAW* weights was not due to chance.

Overall, the results demonstrate that, although both baselines (*LDSD* & *Resim*) and their weighted variations (*WLDSD & WResim*) calculate the semantic distance between resources using the same underlying techniques, our approaches that weight links differentially provide increased accuracy. Also, weighting links using the *RSLAW* approach based on their association with specific classes of resources enables us to identify, and incorporate, latent semantic correlations between links and entities. *WLDSD* and *WResim* demonstrate that links play different roles and should be exploited in any semantic relatedness process for further accurate results.

7.3.1 Example

Figure 12 provides an example of this differential treatment of link types extracted from our experiment results. In this example, *Christina Aguilera* is both directly and indirectly linked to

Sasha Allen and to *Cher* by indirect links only. The baseline *LDSD* approach considers *Cher* is more related to *Christina Aguilera* than *Sasha Allen*; however, our *WLDSD-RSLAW* approach considers *Sasha Allen* to be more related to *Christina Aguilera* than *Cher*. Even though both *Christina Aguilera* and *Cher* have been guests at *The Tonight Show with Jay Leno* twice (through indirect links of type *guests*), their appearance should not be used as an evidence for their relatedness since any famous person can appear at this type of show including politicians who clearly have less association with musical artists than other artists. The same concept applies to the other links between *Christina Aguilera* and *Cher* as most of these links are associated with other resources classes. Although the *LDSD* approach treats all links equally, our *WLDSD-RSLAW* approach differentiates between them based upon their association with musical artists within the entire dataset. In this case, the link types, *associatedMusicalArtist* and *associatedBand*, are highly correlated with the resource class *MusicalArtist* to which *Christina Aguilera*, *Sasha Allen*, and *Cher* belong whereas link types *guests*, *seeAlso*, *extra* are associated with several other resources classes.

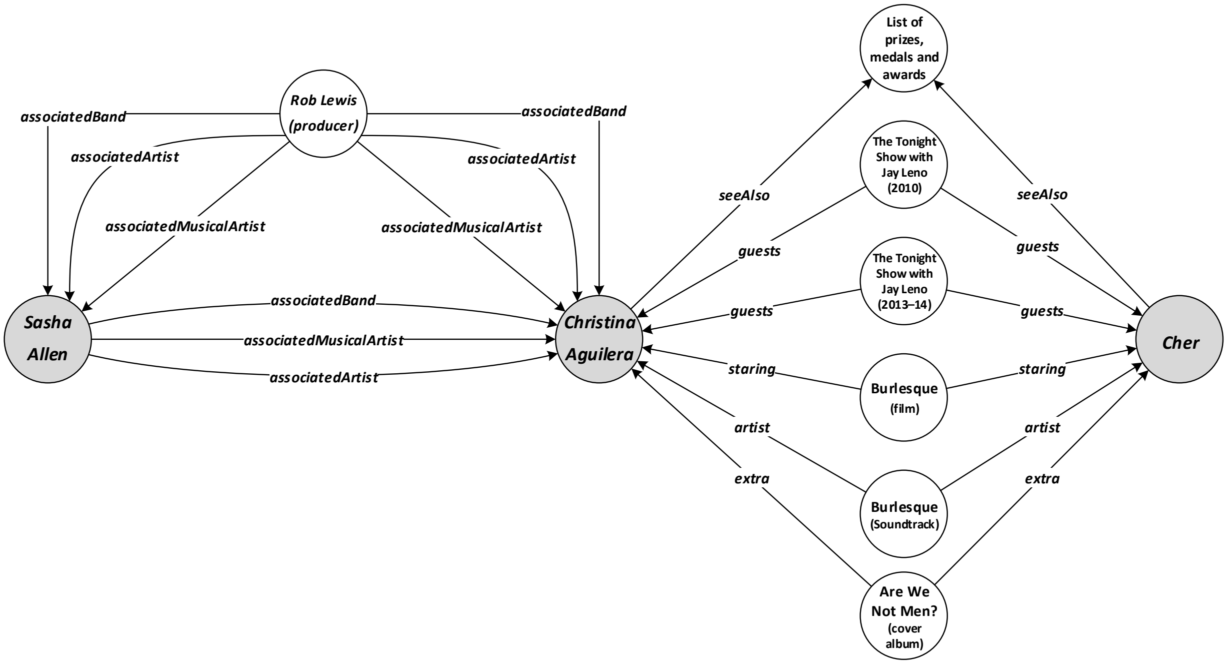


Figure 12: Link differential treatment example

7.4 Experiment 2: Effects of Typeless Links in Semantic Distance

In this experiment, we assess the significance of ignoring link type when calculating the indirect connection (*IC*) part of semantic distance approaches. We compare both baselines, *LDSD* and *Resim*, to their typeless variations. Table 5 shows the results of the experiment using the F_1 score and MRR metrics. The F_1 score values are presented at different ranked results cutoffs, i.e., 5, 10, and 20. In this table, the best results are shown in bold.

Table 5: Experiment 2 results for typeless approaches vs baselines

	Jaccard	*LDSD*	*TLDSD*	*Resim*	*TResim*
MRR	0.010	0.028	0.022	**0.037**	0.024
Precision@5	0.009	0.031	0.008	**0.049**	0.015
Precision@10	0.008	0.033	0.014	**0.040**	0.021
Precision@20	0.008	0.029	0.028	**0.032**	0.030
Recall@5	0.009	0.031	0.008	**0.049**	0.025
Recall@10	0.017	0.065	0.028	**0.079**	0.033
Recall@20	0.029	0.115	0.112	**0.128**	0.097
F1@5	0.009	0.031	0.008	**0.049**	0.019
F_1@10	0.011	0.044	0.019	**0.053**	0.026
F_1@20	0.012	0.046	0.045	**0.051**	0.046

As seen in Table 5, the typeless variations of both baselines underperform their original versions according to all metrics (F_1 and MRR), and this result was statistically significant ($p<0.05$) based on a paired student t-test. The MRR score of the *TLDSD* approach was 0.022 versus 0.028 for the original *LDSD* approach and it was 0.024 for *TResim* versus 0.037 for the original *Resim*. The F_1 score also confirms the MRR metric results with a score of 0.008 for *TLDSD* for the top five results versus a score of 0.031 for the original *LDSD* whereas it was 0.019 for *TResim* versus 0.049 for *Resim*. In particular, the F_1 score at the top ten results for *TLDSD* was 0.019 versus 0.044 for *LDSD*, and it was 0.026 for *TResim* versus 0.053 for *Resim*. Similarly, The F_1 score at the top twenty results for *TLDSD* was 0.045 versus 0.046 for *LDSD*, and it was 0.046 for *TResim* versus 0.051 for *Resim*. Figure 13 displays the MRR values of Table 5 graphically, and Figure 14 displays the three F_1 score results from Table 5 graphically. From these figures, it is clear that the two original approaches outperform both typeless variations and, once again, *Resim* outperforms *LDSD*.

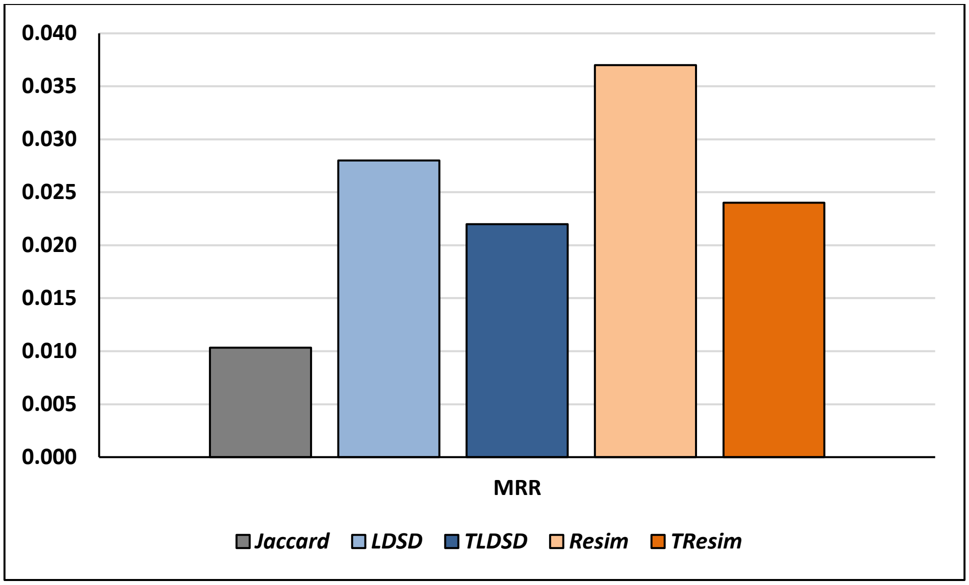


Figure 13: MRR scores for typeless approaches vs baselines

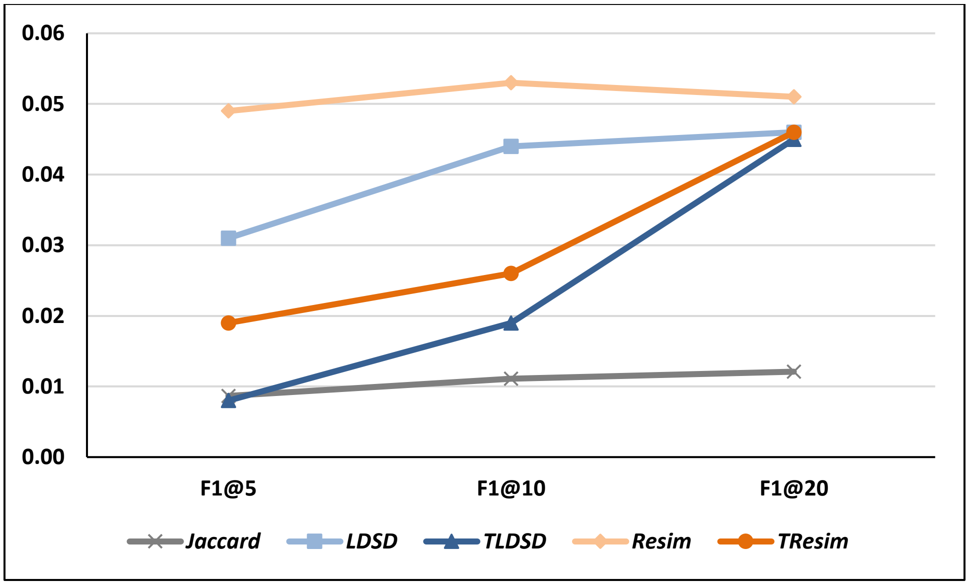


Figure 14: F_1 scores at different ranked results cutoffs for typeless approaches vs baselines

7.5 Experiment 3: Effects of Weighting Typeless Links in Semantic Distance

In this experiment, we investigate the implication of differentially weighting link types in the case of typeless indirect connectivity on semantic distance approaches. This work allows us to find the middle ground between differentiating link based on their importance for recommendation as in *WLDSD* and *WResim* in addition to treating all link types identically in the typeless variations of *LDSD* and *Resim*. Because *RSLAW* was the best performing link weighting approach in our first experiment, we evaluate both baselines, *LDSD* and *Resim*, to their weighted typeless variations using the *RSLAW* weighting method only. Table 6 shows the results of the experiment using the F_1 score and MRR metrics. The F_1 score values are presented at different ranked results cutoffs, i.e., 5, 10, and 20. In this table, the best results are shown in bold.

Table 6: Experiment 3 results for weighted typeless approaches vs baselines

	Jaccard	*LDSD*	*WTLDSD*	*Resim*	*WTResim*
MRR	0.010	0.028	**0.043**	0.037	0.041
Precision@5	0.009	0.031	0.050	0.049	**0.055**
Precision@10	0.008	0.033	**0.049**	0.040	0.041
Precision@20	0.008	0.029	**0.040**	0.032	0.034
Recall@5	0.009	0.031	0.050	0.049	**0.055**
Recall@10	0.017	0.065	**0.097**	0.079	0.083
Recall@20	0.029	0.115	**0.159**	0.128	0.134
F1@5	0.009	0.031	0.050	0.049	**0.055**
F_1@10	0.011	0.044	**0.065**	0.053	0.055
F_1@20	0.012	0.046	**0.063**	0.051	0.054

As seen in Table 6, both weighted typeless variations of our baselines outperformed their original variations in all metrics (F_1 and MRR), and this result was statistically significant

($p<0.05$) based on a paired student t-test. The MRR score of the *WTLDSD* approach was 0.043 versus 0.028 for the original *LDSD* approach while it was 0.041 for *WTResim* versus 0.037 for *Resim*.

The F_1 score also confirms the MRR metric results with a score of 0.050 for *WTLDSD* for the top five results versus a score of 0.031 for the original *LDSD* whereas it was 0.055 for *WTResim* versus 0.049 for *Resim*. These results also hold at other results cutoff points as displayed in Figure 16. In particular, the F_1 score at the top ten results for *WTLDSD* was 0.065 versus 0.044 for *LDSD*, and it was 0.055 for *WTResim* versus 0.053 for *Resim*. Similarly, The F_1 score at the top twenty results for *WTLDSD* was 0.063 versus 0.046 for *LDSD*, and it was 0.054 for *WTResim* versus 0.051 for *Resim*. Figures 15 and 16 show these results graphically, plotting the MRR scores and F1 scores, respectively.

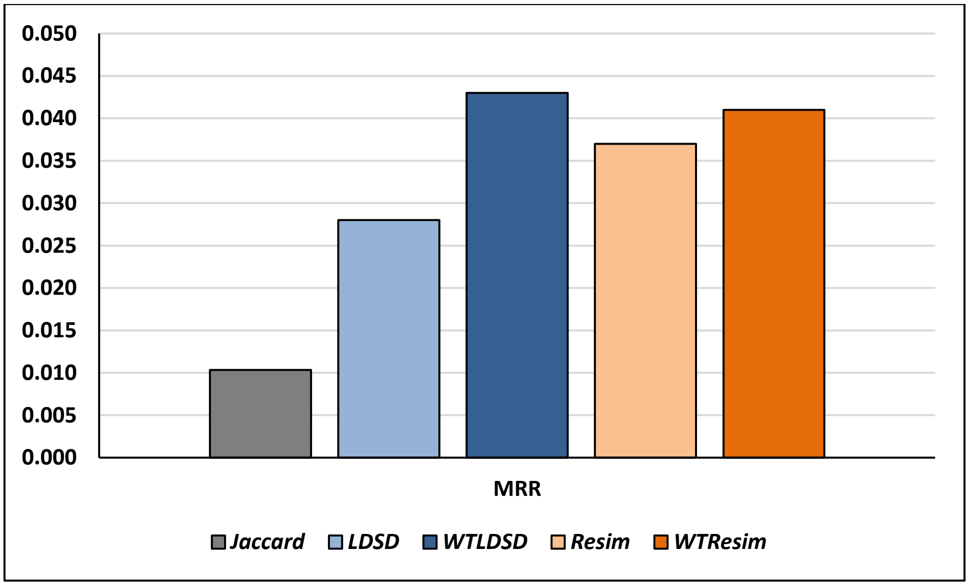


Figure 15: MRR scores for weighted typeless approaches vs baselines

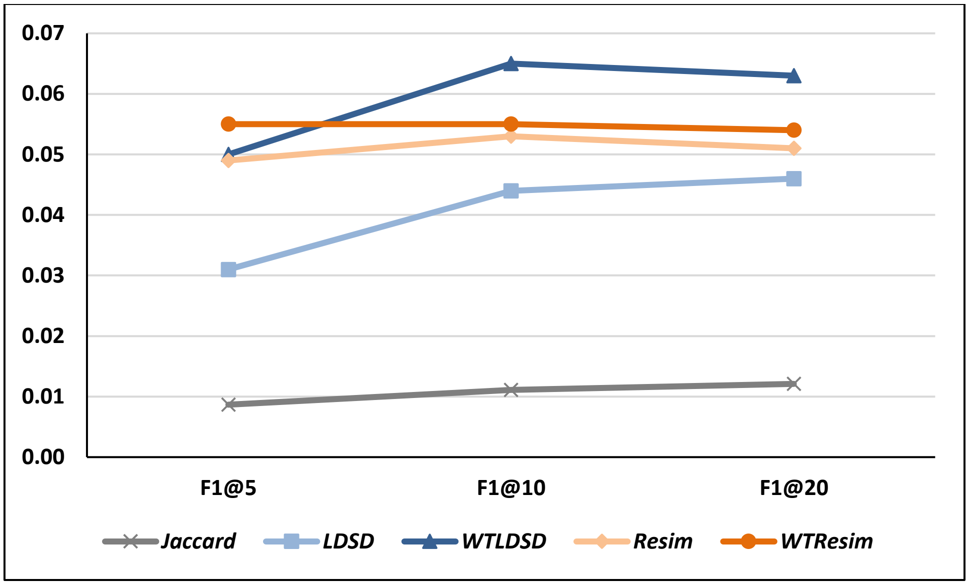


Figure 16: F$_1$ scores at different ranked results cutoffs for weighted typeless approaches vs baselines

The results also show that adding weights to *LDSD* had a more dramatic effect than adding weights to *Resim*. This is because although *LDSD* considers connections only between resources linked via direct or indirect connections, whereas *Resim* adds another similarity calculations incase direct or indirect connections do not exist by calculating the similarities between resources' property vectors.

Finally, we can see that although *Resim* outperforms *LDSD*, with weights included, weighted *LDSD* outperforms weighted *Resim*. This indicates that, where direct and indirect connections are nonexistent, our weighted similarity calculation outperforms the similarity method of properties in *Resim*. These results show the significance of exploiting link weights in semantic distance computation as the accuracy of the recommender system increases even when typeless indirect connectivity is used. All things considered, the weighted typeless approaches outperform also the weighted approaches yielding in the highest accurate approach so far, *WTLDSD*.

7.6 Experiment 4: Effects of Propagating Semantic Distances

Our fourth and final experiment evaluates the effect of propagating semantic distances through the LOD graph from the current restriction of being no more than one link away. We ran this experiment on both baselines *LDSD* and *Resim* in addition to their weighted typeless variations (*WTLDSD* and *WTResim*). Table 7 shows the results of the experiment using the precision, recall, F_1 score and MRR metrics. The precision, recall, and F_1 score values are presented at different ranked results cutoffs, i.e., 5, 10, and 20. In this table, the best results are shown in bold.

Table 7: Experiment 4 results for propagated approaches vs others

	Jaccard	*LDSD*	*pLDSD*	*WTLDSD*	*pWTLDSD*	*Resim*	*pResim*	*WTResim*	*pWTResim*
MRR	0.010	0.028	0.044	0.043	**0.050**	0.037	0.032	0.041	0.032
Precision@5	0.009	0.031	0.051	0.050	**0.065**	0.049	0.034	0.055	0.036
Precision@10	0.008	0.033	0.044	0.049	**0.056**	0.040	0.039	0.041	0.041
Precision@20	0.008	0.029	0.037	0.040	**0.050**	0.032	0.037	0.034	0.036
Recall@5	0.009	0.031	0.051	0.050	**0.065**	0.049	0.034	0.055	0.036
Recall@10	0.017	0.065	0.088	0.097	**0.113**	0.079	0.079	0.083	0.083
Recall@20	0.029	0.115	0.146	0.159	**0.199**	0.128	0.148	0.134	0.142
F1@5	0.009	0.031	0.051	0.050	**0.065**	0.049	0.034	0.055	0.036
F1@10	0.011	0.044	0.059	0.065	**0.075**	0.053	0.052	0.055	0.055
F_1@20	0.012	0.046	0.058	0.063	**0.079**	0.051	0.059	0.054	0.057

As the results show, our propagated LDSD-based approaches (*pLDSD* and *pWTLDSD*) outperformed their corresponding baselines for LDSD-based approaches (*LDSD* and *WTLDSD*) in all metrics (F_1 and MRR). Figure 17 displays the MRR scores for all the approaches in this experiment. The MRR score of the *pLDSD* approach was 0.044 versus 0.028 for the original *LDSD*, an improvement of 57%, while it was 0.050 for *pWTLDSD* versus 0.043 for *WTLDSD*, an improvement of 16%. The F_1 score also confirms the MRR metric results with a score of 0.051 for *pLDSD* for the top five results versus a score of 0.031 for the original *LDSD* whereas it was 0.065 for *pWTLDSD* versus 0.050 for *WTLDSD*. These results also hold at other results cutoff points (@10 and @20).

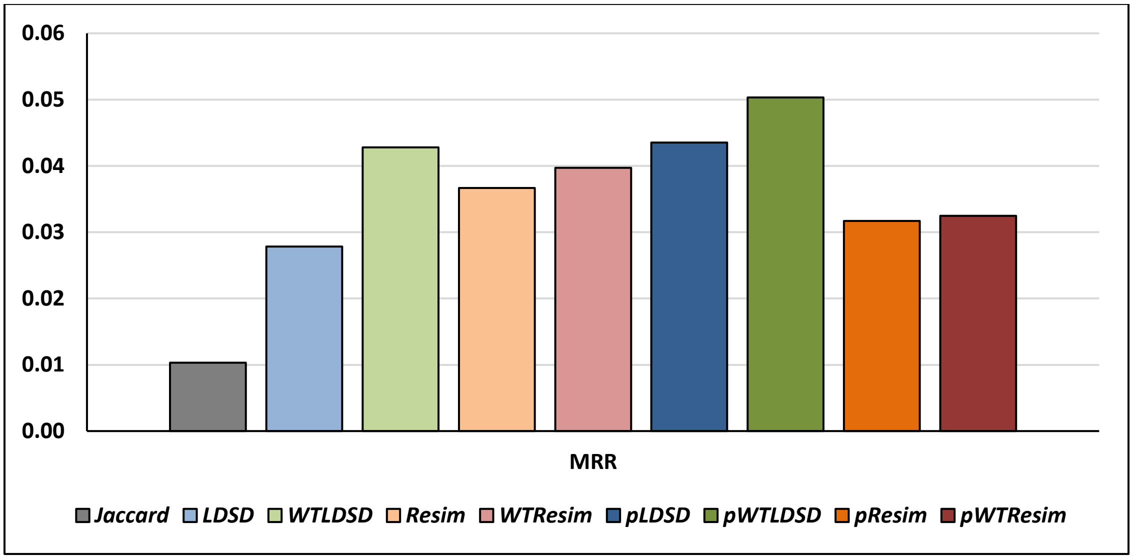


Figure 17: MRR scores for propagated approaches vs others

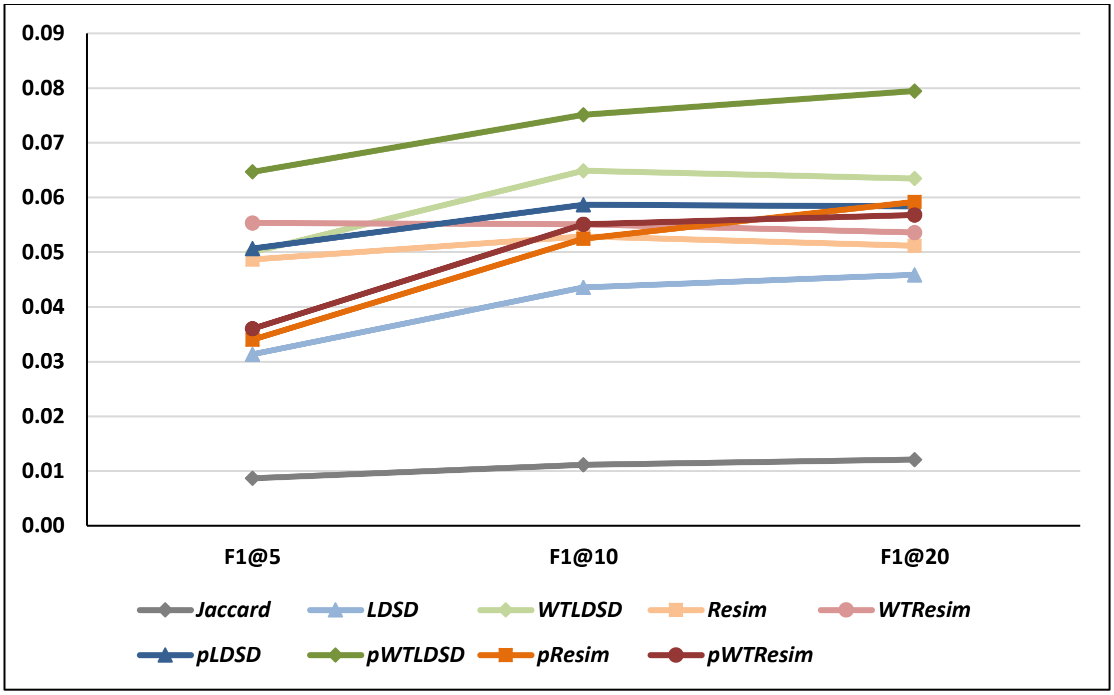


Figure 18: F1 scores at different ranked results cutoffs for propagated approaches vs others

On the other hand, the propagated variations of Resim-based approaches (*pResim* and *pWTResim*) fall behind their corresponding baselines (*Resim* and *WTResim*). The MRR score of the *pResim* approach was 0.032 versus 0.037 for the original *Resim*, and it was 0.032 for *pWTResim* versus 0.041 for *WTResim*. The F_1 score for the top five results was 0.034 for *pResim* versus 0.049 for *Resim*, and it was 0.036 for *pWTResim* versus 0.035 for *WTResim*. These results also hold at other results cutoff points (@10 and @20). Even though our propagated approach did not perform well in *Resim*-based semantic distances, it only fell behind for the top five results while it was comparable at the top ten results and was better at the top 20 results as seen in Figure 18.

The *pWTLDSD* gained the highest accuracy among all the approaches in this document with an improvement of 78% over our first baseline in this document, *LDSD*, and 35% over *Resim*. It also gained an improvement of 16% over our previous best performing approach

(*WTLDSD*). Overall, these results show that propagating semantic distances beyond one hub resources improves the accuracy of LOD-based recommender systems.

Recommender systems are not only evaluated by their accuracy; they can also be evaluated by other criteria including their coverage. Thus, it is important to note that our propagated approach increased the coverage. According to [56], the coverage of a recommender system is defined as the percentage of the dataset that the system is able include its recommendation results. As a reminder, semantic distance calculations on a pair of resources produce results on a scale of 0 (no distance apart, therefore completely identical) to 1 (as far away as possible, therefore completely unrelated). Thus, related results are defined as all resources with a semantic distance less than 1 whereas non-related resources are those with a semantic distance of exactly 1. Table 8 shows the coverage of the propagated approaches vs others. The coverage here is defined as the average number of related results per resource as follows:

$$Coverage = \frac{\sum_{r_i}\left(\frac{\sum_{r_j \in R(r_i)} 1}{n}\right)}{n} \times 100$$

where *n* is the number of resources in the dataset, and *R(*r_i*)* is the set of resources that have semantic distances less than 1 from r_i.

These results demonstrate clearly that the coverage of each approach increases when the propagated approach is applied with a maximum increase of 81% (*pWTLDSD vs WTLDSD*) and a minimum increase of 33% (*pResim* vs *Resim*). As a result of this coverage increment, the recommender system has access to more possible related resources that may result in an enhanced novelty of the recommendation results.

Table 8: The coverage of the propagated approaches vs others

	LDSD	*pLDSD*	*WTLDSD*	*pWTLDSD*	*Resim*	*pResim*	*WTResim*	*pWTResim*
Coverage	10%	85%	9%	90%	61%	94%	60%	95%

7.7 Discussion

Figure 19 summarizes the overall performance for all approaches. Because *RSLAW* weights outperformed *ITW* weights, each of the weighted approaches in this figure is based on *RSLAW* weights. From this we can see that the best performing approach is our *pWTLDSD* that presents an improvement of 79% over *LDSD* and 36% over *Resim*. The *pWTLDSD* combines links weighting approach in typeless indirect connectivity with propagated semantic distances. All the weighted approaches (*WLDSD*, *WTLDSD*, *WResim*, and *WTResim*) gained better accuracy than their corresponding non-weighted variations (*LDSD*, *TLDSD*, *Resim*, and *TResim*); showing the significance of distinguishing links based on the level of relatedness between resources indicated by each. Moreover, propagating semantic distances beyond one hub resources does not only result in an improved accuracy as in *pLDSD* and *pWTLDSD*, it also shows that propagating semantic distances beyond one hub resources improves the coverage of LOD-based recommender systems.

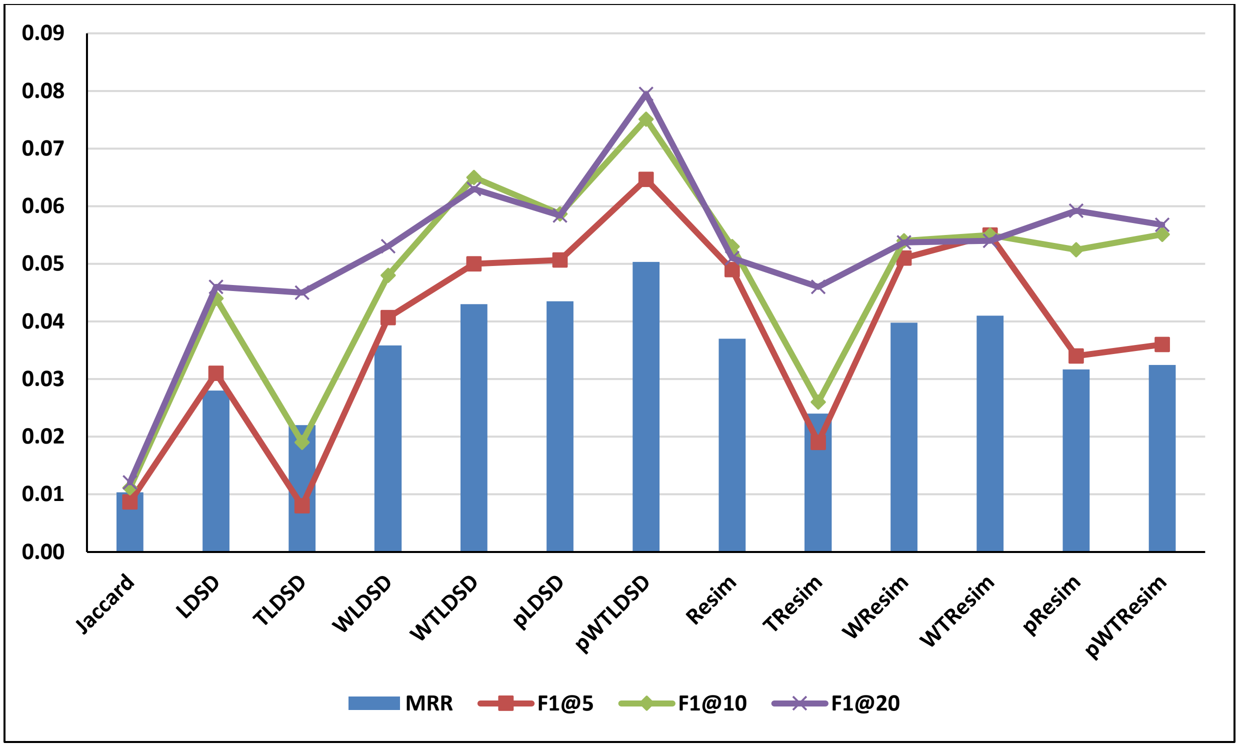


Figure 19: All approaches overview

One lesson learned in this study is that we need to manage the time complexity of current semantic distance approaches. In particular, propagating weights throughout a network leads to combinatorial explosion. Since we executed our experiments in a live public instance of *DBpedia* that contains over 5 million resources with more than 397 million properties, the n^2 semantic distance computations between resources took a long time. Our experiments required an average of 8 days to complete, primarily because of the slow response of the *DBpedia* server. Several studies [57] [58] point to this problem and suggest enhanced implementations of current software of LOD engines. Trying to tackle this challenge, we created a local replica of *DBpedia* in our machines, but we did not experience a substantial speedup compared to the live public server. A previous study [9] suggested using a reduced LOD dataset by creating a compact LOD dataset that contains only resources to be recommended; nonetheless, we preferred to use the complete live public version to emulate real life scenarios. [9] recommended also breaking SPARQL queries into several smaller queries instead of one larger complete query to increase the response time; a tactic we adopted. One additional tactic we applied is caching, so every SPARQL request is cached locally. In a nutshell, efficient LOD engines are a necessity if we are to effectively and efficiently utilize the huge amount of publicly available Linked Open Data. In addition, future semantic distance studies should take computation efficiency into consideration.

8 Conclusion

8.1 Summary

The rise of Linked Open Data has encouraged researchers to exploit it in recommender systems through identifying the relatedness between resources in LOD. One approach is to compute the semantic distance between resources to recognize their relatedness. In this document, we showed that different types of resources links hold different values for relatedness calculations, and we exploited this observation to introduce improved weighted resource semantic relatedness measures that is more accurate than current approaches. In our methods, we distinguished links based their type by introducing a weighting factor for every link, and then we calculated this weight based on the association rate of the link type within a specific resource class. Also, we introduce a new approach that expands the coverage of current semantic distance approaches to include additional resources. We employ an all-pair shortest path algorithm, namely, the well-known Floyd-Warshall algorithm, to efficiently compute semantic distances based on resources more than beyond one or two links away.

To verify our observations, we conducted an experiment in the music domain, and its results showed that the best performing approach is our *pWTLDSD* that presents an improvement of 79% over *LDSD* and 36% over *Resim*. The *pWTLDSD* combines links weighting approach in typeless indirect connectivity with propagated semantic distances. All the weighted approaches (*WLDSD*, *WTLDSD*, *WResim*, and *WTResim*) gained better accuracy than their corresponding non-weighted variations (*LDSD*, *TLDSD*, *Resim*, and *TResim*); showing the significance of distinguishing links based on the level of relatedness between resources indicated by each. Moreover, propagating semantic distances beyond one hub resources does not only result in an improved accuracy as in *pLDSD* and *pWTLDSD*, it also shows that propagating semantic distances beyond one hub resources improves the coverage of LOD-based recommender systems.

8.2 Future Work

In future, we will explore different ways to calculate the links weights. One possible approach is to combine link type nature with path-based normalization to achieve higher relatedness accuracy. Furthermore, we will analyze the effects of our proposed approaches on different domains such as books and movies as well as to perform cross-domain recommendations. Moreover, we will investigate improving the efficiency of LOD similarity measures in term of time complexity in order to achieve efficient similarity approaches with comparable accuracy and coverage.

9 References

[1] P. Cremonesi, A. Tripodi and R. Turrin, "Cross-Domain Recommender Systems," in *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, Washington, DC, USA, 2011.

[2] T. Berners-Lee, "Linked Data," 27 07 2006. [Online]. Available: https://www.w3.org/DesignIssues/LinkedData.html. [Accessed 07 04 2016].

[3] M. Schmachtenberg, C. Bizer and H. Paulheim, "Adoption of the linked data best practices in different topical domains," in *International Semantic Web Conference*, Riva del Garda, Italy, 2014.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*, Springer Berlin Heidelberg, 2007, pp. 722-735.

[5] D. Damljanovic, M. Stankovic and P. Laublet, "Linked data-based concept recommendation: Comparison of different methods in open innovation scenario," *The Semantic Web: Research and Applications*, pp. 24-38, 2012.

[6] T. Di Noia and V. Ostuni, "Recommender Systems and Linked Open Data," in *Reasoning Web. Web Logic Rules*, vol. 9203, W. a. P. A. Faber, Ed., Springer International Publishing, 2015, pp. 88-113.

[7] C. Musto, F. Narducci, P. Lops, M. De Gemmis and G. Semeraro, "ExpLOD: A Framework for Explaining Recommendations based on the Linked Open Data Cloud," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016.

[8] C. Figueroa, I. Vagliano, O. R. Rocha and M. Morisio, "A systematic literature review of Linked Data-based recommender systems," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 17, pp. 4659-4684, 2015.

[9] A. Passant, "Measuring Semantic Distance on Linking Data and Using it for Resources Recommendations," in *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*, 2010.

[10] G. Piao, S. showkat Ara and J. G. Breslin, "Computing the Semantic Similarity of Resources in DBpedia for Recommendation Purposes," in *Joint International Semantic Technology Conference*, 2015.

[11] G. Piao and J. G. Breslin, "Measuring semantic distance for linked open data-enabled recommender systems," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, 2016.

[12] F. Ricci, L. Rokach and B. Shapira, Introduction to recommender systems handbook, Springer US, 2011.

[13] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction,* vol. 12, no. 4, pp. 331-370, 2002.

[14] 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,* vol. 17, no. 6, pp. 734-749, 2005.

[15] R. Burke, "Hybrid web recommender systems," in *The adaptive web*, Springer, 2007, pp. 377-408.

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

[17] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence,* p. 4, 2009.

[18] D. Parra and S. Sahebi, "Recommender systems: Sources of knowledge and evaluation metrics," in *Advanced Techniques in Web Intelligence-2*, Springer Berlin Heidelberg, 2013, pp. 149-175.

[19] I. Fernandez-Tobias, I. Cantador, M. Kaminskas and F. Ricci, "Cross-domain recommender systems: A survey of the state of the art," *Spanish Conference on Information Retrieval,* 2012.

[20] I. Cantador and P. Cremonesi, "Tutorial on Cross-domain Recommender Systems," in *Proceedings of the 8th ACM Conference on Recommender Systems*, New York, NY, USA, 2014.

[21] T. Gottron and S. Staab, "Linked Open Data," in *Encyclopedia of Social Network Analysis and Mining,* Springer New York, 2014, pp. 811-813.

[22] T. Berners-Lee, "Uniform Resource Identifier (URI): Generic Syntax," Internet Engineering Task Force, 01 01 2005. [Online]. Available: https://tools.ietf.org/html/rfc3986. [Accessed 07 04 2016].

[23] L. Yu, "Linked open data," in *A Developer's Guide to the Semantic Web*, Springer Berlin Heidelberg, 2011, pp. 409-466.

[24] L. F. Sikos, "Linked Open Data," in *Mastering Structured Data on the Semantic Web*, Apress, 2015, pp. 59-77.

[25] K. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008.

[26] K. Bollacker, P. Tufts, T. Pierce and R. Cook, "A platform for scalable, collaborative, structured information integration," in *Intl. Workshop on Information Integration on the Web (IIWeb'07)*, 2007.

[27] A. Singhal, May 2012. [Online]. Available: https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html.

[28] P. Jain, P. Hitzler, A. P. Sheth, K. Verma and P. Z. Yeh, "Ontology alignment for linked open data," in *International Semantic Web Conference*, 2010.

[29] G. Kobilarov, T. Scott, Y. Raimond, S. Oliver, C. Sizemore, M. Smethurst, C. Bizer and R. Lee, "Media meets semantic web–how the BBC uses DBpedia and linked data to make connections," in *The semantic web: research and applications*, Springer Berlin Heidelberg, 2009, pp. 723-737.

[30] A. Passant, "Dbrec: Music Recommendations Using DBpedia," in *Proceedings of the 9th International Semantic Web Conference on The Semantic Web - Volume Part II*, Springer-Verlag, 2010, pp. 209-224.

[31] J. P. Leal, V. Rodrigues and R. Queirós, "Computing semantic relatedness using dbpedia," in *1st Symposium on Languages, Applications and Technologies (SLATE'12)*, 2012.

[32] J. P. Leal, "Using proximity to compute semantic relatedness in RDF graphs," *Computer Science and Information Systems*, vol. 10, no. 4, pp. 1727-1746, 2013.

[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*, 2015.

[34] I. Fernández-Tobías, I. Cantador, M. Kaminskas and F. Ricci, "A generic semantic-based framework for cross-domain recommendation," in *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, 2011.

[35] 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*, 2012.

[36] 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*, 2012.

[37] T. Di Noia, R. Mirizzi, V. C. Ostuni, D. Romito and M. Zanker, "Linked open data to support content-based recommender systems," in *Proceedings of the 8th International Conference on Semantic Systems*, 2012.

[38] V. C. Ostuni, T. Di Noia, R. Mirizzi, D. Romito and E. Di Sciascio, "Cinemappy: a Context-aware Mobile App for Movie Recommendations boosted by DBpedia.," in *SeRSy*, 2012.

[39] 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*, 2013.

[40] V. C. Ostuni, T. Di Noia, R. Mirizzi and E. Di Sciascio, "A linked data recommender system using a neighborhood-based graph kernel," in *E-Commerce and Web Technologies*, Springer International Publishing, 2014, pp. 89-100.

[41] P. T. Nguyen, P. Tomeo, T. Di Noia and E. Di Sciascio, "Content-Based Recommendations via DBpedia and Freebase: A Case Study in the Music Domain," in *The Semantic Web--ISWC 2015*, Springer International Publishing, 2015, pp. 605-621.

[42] R. Meymandpour and J. Davis, "Enhancing Recommender Systems Using Linked Open Data-Based Semantic Analysis of Items," in *3rd Australasian Web Conference (AWC 2015)*, 2015.

[43] B. Heitmann and C. Hayes, "Using Linked Data to Build Open, Collaborative Recommender Systems.," in *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*, 2010.

[44] B. Heitmann, "An open framework for multi-source, cross-domain personalisation with semantic interest graphs," in *Proceedings of the sixth ACM conference on Recommender systems*, 2012.

[45] C. Musto, P. Basile, P. Lops, M. de Gemmis and G. Semeraro, "Linked Open Data-enabled Strategies for Top-N Recommendations," *CBRecSys 2014*, p. 49, 2014.

[46] L. Peska and P. Vojtas, "Using linked open data to improve recommending on e-commerce," in *2nd International Workshop on Semantic Technologies meet Recommender Systems & Big Data (SeRSy 2013)*, 2013.

[47] L. Peska and P. Vojtas, "Using Linked Open Data in Recommender Systems," in *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, 2015.

[48] Y. Kabutoya, R. Sumi, T. Iwata, T. Uchiyama and T. Uchiyama, "A Topic Model for Recommending Movies via Linked Open Data," in *International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 2012.

[49] G. Salton and M. J. McGill, Introduction to Modern Information Retrieval, New York, NY: McGraw-Hill, Inc, 1983.

[50] R. W. Floyd, "Algorithm 97: Shortest Path," *Communications of the ACM,* vol. 5, no. 6, p. 345, 1962.

[51] S. Alfarhood, L. Kevin and S. Gauch, "PLDSD: Propagated Linked Data Semantic Distance," in *Preceedings of the The 26th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE-2017),* Poznan, Poland, 2017.

[52] P. Jaccard, "Etude de la distribution florale dans une portion des Alpes et du Jura," *Bulletin de la Societe Vaudoise des Sciences Naturelles* , vol. 37, no. 142, p. 547–579, 1901.

[53] H. Steck, "Evaluation of recommendations: rating-prediction and ranking," in *Proceedings of the 7th ACM conference on Recommender systems*, 2013.

[54] C. J. V. Rijsbergen, Information Retrieval, London, 1979.

[55] N. Craswell, "Mean Reciprocal Rank," in *Encyclopedia of Database Systems*, L. LIU and M. T. OZSU, Eds., Boston, MA, Springer, 2009, pp. 1703-1703.

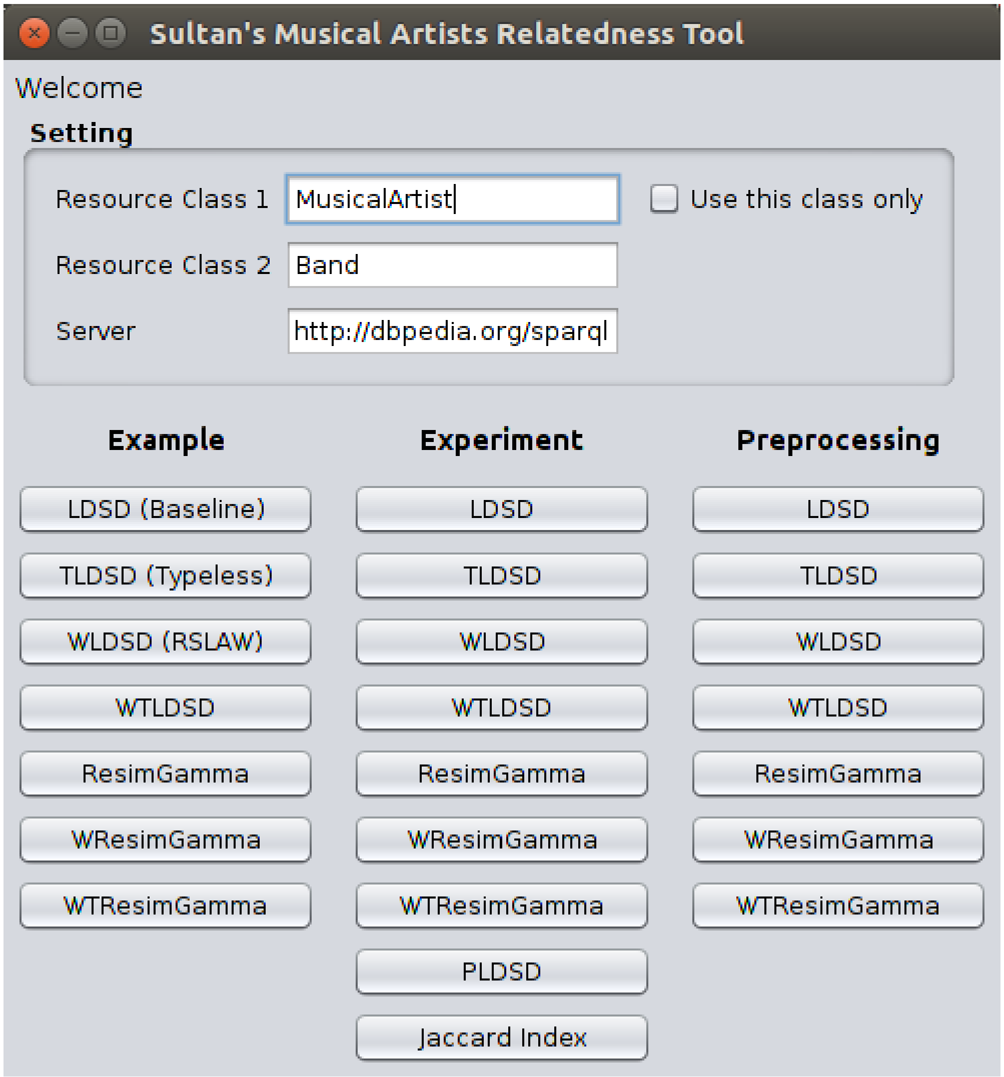
[56] J. L. Herlocker, J. A. Konstan, L. G. Terveen and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS),* vol. 22, no. 1, pp. 5-53, 2004.

[57] A. Dessi, A. Maxia, M. Atzori and C. Zaniolo, "Supporting semantic web search and structured queries on mobile devices," in *Proceedings of the 3rd International Workshop on Semantic Search Over the Web*, 2013.

[58] C. Bizer and A. Schultz, "Benchmarking the performance of storage systems that expose SPARQL endpoints," in *Proceedings of the ISWC Workshop on Scalable Semantic Web Knowledgebase*, 2008.

10 Appendix

10.1 Screenshots



Sultan's Musical Artists Relatedness Tool

Welcome

Setting

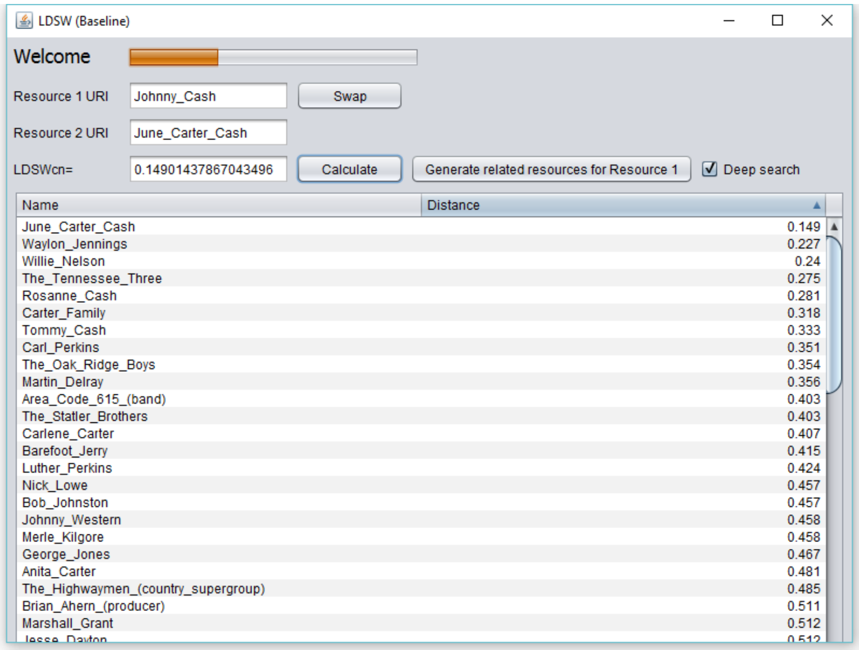
Resource Class 1 MusicalArtist

☐ Use this class only

Resource Class 2 Band

Server http://dbpedia.org/sparql

Figure 20: Experiment main window



Resource 1 URI: Johnny_Cash

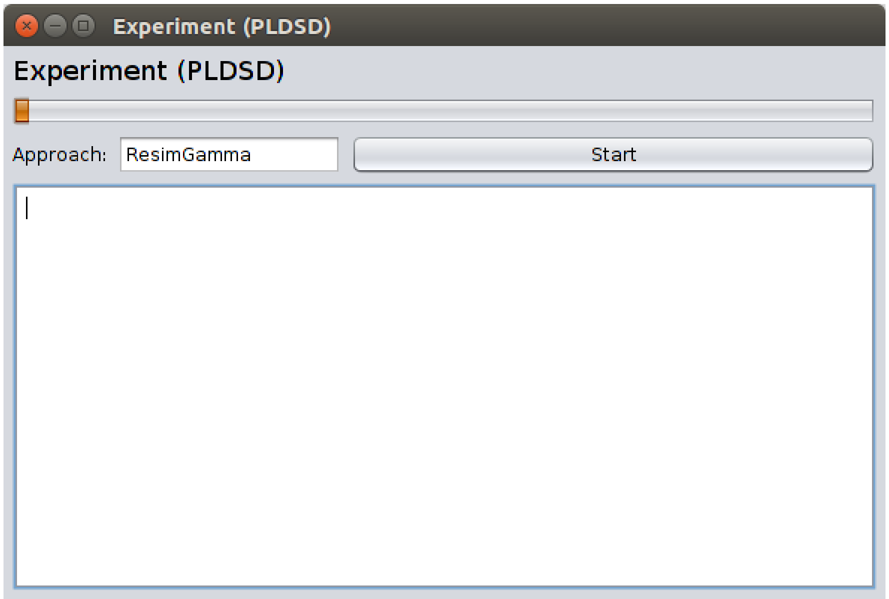
Resource 2 URI: June_Carter_Cash

LDSWcn= 0.14901437867043496

☑ Deep search

Name	Distance
June_Carter_Cash	0.149
Waylon_Jennings	0.227
Willie_Nelson	0.24
The_Tennessee_Three	0.275
Rosanne_Cash	0.281
Carter_Family	0.318
Tommy_Cash	0.333
Carl_Perkins	0.351
The_Oak_Ridge_Boys	0.354
Martin_Delray	0.356
Area_Code_615_(band)	0.403
The_Statler_Brothers	0.403
Carlene_Carter	0.407
Barefoot_Jerry	0.415
Luther_Perkins	0.424
Nick_Lowe	0.457
Bob_Johnston	0.457
Johnny_Western	0.458
Merle_Kilgore	0.458
George_Jones	0.467
Anita_Carter	0.481
The_Highwaymen_(country_supergroup)	0.485
Brian_Ahern_(producer)	0.511
Marshall_Grant	0.512
Jesse_Dayton	0.512

Figure 21: LDSD approach calculation window



Approach: ResimGamma

Figure 22: PLDSD experiment window