Published in final edited form as:

Comput Biol Med. 2017 August 01; 87: 217–229. doi:10.1016/j.compbimed.2017.05.026.

Supporting Inter-topic Entity Search for Biomedical Linked Data Based on Heterogeneous Relationships

Nansu Zong^{1,*}, Sungin Lee², Jinhyun Ahn², and Hong-Gee Kim²

¹Department of Biomedical Informatics, School of Medicine, University of California, San Diego, USA

²Biomedical Knowledge Engineering Laboratory, School of Dentistry, Seoul National University, Korea

Abstract

Objective—The keyword-based entity search restricts search space based on the preference of search. When given keywords and preferences are not related to the same biomedical topic, existing biomedical Linked Data search engines fail to deliver satisfactory results. This research aims to tackle this issue by supporting an inter-topic search—improving search with inputs, keywords and preferences, under different topics.

Methods—This study developed an effective algorithm in which the relations between biomedical entities were used in tandem with a keyword-based entity search, Siren. The algorithm, PERank, which is an adaptation of Personalized PageRank (PPR), uses a pair of input: (1) search preferences, and (2) entities from a keyword-based entity search with a keyword query, to formalize the search results on-the-fly based on the index of the precomputed Individual Personalized PageRank Vectors (IPPVs).

Results—Our experiments were performed over ten linked life datasets for two query sets, one with keyword-preference topic correspondence (intra-topic search), and the other without (inter-topic search). The experiments showed that the proposed method achieved better search results, for example a 14% increase in precision for the inter-topic search than the baseline keyword-based search engine.

Conclusion—The proposed method improved the keyword-based biomedical entity search by supporting the inter-topic search without affecting the intra-topic search based on the relations between different entities.

Keywords

Keyword-based Entity Search; Inter-topic Search; Personalized PageRank; Biomedical Linked Data

Correspondence to: Nansu Zong, Department of Biomedical Informatics School of Medicine University of California, San Diego, USA, nazong@ucsd.edu.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

1. Introduction

As biomedical data increases, Resource Description Framework (RDF) (<http://www.w3.org/RDF/>) has been adopted as the standard data format in Linked Data for interconnection, integration, and reuse of published biomedical data [1]. Biomedical Linked Data has now garnered more than 10 billion links (<http://linkedlifedata.com/sources.html>) connecting entities in diverse topics, including medicine, drug, symptom, gene, and others.

The keyword-based search, also known as text-based search, adapts the existing information retrieval models to provide scalable indexing capabilities as well as a user-friendly experience for an entity search [2, 3]. With a set of keywords as the input query, retrieved entities are sorted with a descending order of the relevancy, while excluding the need to understand the schema of data [2]. The relevancy is computed with two types of models [4, 5]: (1) query-dependent models, such as Siren [6], RareRank [7], ObjectRank [8]; and (2) query-independent models, such as ReCibRabk [9], Swoogle [10]. In order to provide better results, two methods are used in general for: (1) query expansions, and (2) filters in advanced search. Query expansion tends to retrieve more research results by enriching the keyword query with expansions, which are the related term(s) (e.g., synonym, hypernym) based on knowledge-bases [11] or Web of document [12], statistical related entities based on co-occurrence [13], or the preferences [14]. Filters use additional input, e.g., search preferences, to narrow down the search space to specific topics to retrieve more precise results. This strategy works for the intra-topic search in which the topics pertaining to the query keywords correspond with the search preference (e.g., searching drugs by name), however it provides dissatisfactory results when the topics are somewhat related but disparate.

Consider the example shown in Fig. 1: when attempting to find information on “drugs” (denoting the preference of search) for the disease “renal tubular acidosis” (the keyword query), the system will restrict the search space into the entities belonging to the designated preferred topic (i.e., drugs) and search for the entities that contain keywords in their descriptions. As a result, the unrelated drugs “aminohippurate”, “torasemide”, and “acyclovir” are returned by a keyword-based entity search engine, Linked Life Data (<http://linkedlifedata.com/search/quick>). In this search, the topics are used to narrow down the data spaces, and only the *is_A* relation is used in the search while other relations between topics are ignored. Therefore, keywords associated with a search preference that inherently requires the use of cross-topic links are not best dealt with by such keyword-based search systems. A system that can search the entities by considering (1) the keywords related to the entity, and (2) the preference related to the class (topic) of the entity, needs to be studied.

To improve the existing biomedical entity search method for strengthening the ability of tackling the inter-topic search, this paper introduces PERank (<https://github.com/zongnansu1982/PERankComputation>) that uses the preference of search to trace the semantic meanings of the links to obtain more precise results. PERank adapts Personalized PageRank (PPR) to compute ranking scores of entities based on a pair of input: (1) a preference of search, and (2) entities returned from an entity search system with a keyword query, such as Siren (<http://rdelbru.github.io/SIREn/>). PERank refactors PPR with, (1) a

customized transition matrix based on the classes and predicates in the schema graph, and (2) a personalized vector based on the entities in the data graph, to generate the Individual Personalized PageRank Vectors (IPPVs), which will be used to calculate a ranking score for each pair of input. PERank was tested with 10 linked life datasets and compared with the baseline search methods in light of various entity search strategies, including query expansion, class and predicate filters. The experiments demonstrate that the entities related to both the keyword and preference can be successfully retrieved by PERank, especially for inter-topic search. For example, in searching “diseases” (preference) that can be treated with “ethinamate” (keyword query), PERank returns two kinds of diseases: “renal tubular acidosis” and “acidosis-osteopetrosis syndrome” (shown in Appendix B), while other methods cannot return related ones. PERank improves the search (e.g., a 14% increase in precision compared to the baseline keyword-based search engine) when the topics of the keywords and preference do not correspond with each other.

The rest of the paper is organized as follows: in Section 2, we introduce the related works of this study. We present the basic idea of PERank in Section 3, and describe the computation of PERank with mathematical definitions in Sections 4. In Section 5, we present our experiment results. Finally, we discuss the limitations and conclude our work in Section 6 and 7.

2. Related works

Entity search, which finds the entities containing a given query in the properties, has been studied for years. Many biomedical platforms provide SPARQL endpoints to search over the data, such as EBI [15], Uniprot [16], Medical Subject Headings (Mesh) (<https://id.nlm.nih.gov/mesh/query>), NCBI2RDF [17], Big Linked Cancer Data [18], Biotea [19]. However, the limitation of triple store and SPARQL query in indexing scalability and the query formalization has driven Information Retrieval approaches to deal with entity search over the past few years [2]. In contrast to the triple store, keyword-based approaches store all the triples about one entity as an RDF document, such as the star-shaped document [6, 20], and return it as a representation of the entity [2]. The similarity between the document and the keywords is calculated by different IR models, such as Vector Space Model (VSM) [6] and the probabilistic model BM25F [2, 21]. Based on these models, there are a few applications to search entities in general domains. Siren [6], one of the most popular entity search frameworks, adapts TF/IDF [22] to search Linked Data based on the framework of Lucene (<http://lucene.apache.org/>) and offers a good compromise between query expressiveness, query processing, and index maintenance. Similarly, Falcons [23], Swoogle [10], and SWSE [9] search Linked Data based on VSM with virtual documents, but sort retrieved entities with different methods. Falcons adapts as the combination of query relevancy and entity popularity, and Swoogle and SWSE adapt variants of PageRank (PR) for ranking. Several studies [2, 21] implemented the BM25F retrieval model and reported good results. In the biomedical domain, Linked Life Data (<http://linkedlifedata.com/>) and Bio2RDF [24] generate linked datasets based on existing biomedical repositories and provide keyword-based search over biomedical entities. These methods focus on searching Linked Data with a keyword query and use preferences as a filter to narrow down the search

space. The semantic relations between the preference and keywords, except for *is_A*, are ignored in these methods.

Based on the keyword-based search, this study proposes PERank, a variant of PPR, to improve search by exploring the semantic relations between the entities to support an inter-topic search with a given preference. PPR originates from the PR algorithm [25], which allows the PR score to be biased towards specific topics. By using the non-uniform personalized vectors (where there are 16 topics in Open Directory Project), Topic Sensitive PageRank (TSPR) teleports more authority to the webpages belonging to a designated topic to increase the importance of these pages for searching the information in such topic [26]. Similarly, the pages can receive the customized rankings based on the “blocks” formed by the same hosts [27]. ObjectRank is the first study that tries to apply PPR to entity-relationship (ER) graph search [8]. There are two notable differences between ObjectRank and the traditional PPR. Firstly, ObjectRank takes predicates for teleporting the authorities into consideration by predefining the weights of links. Secondly, ObjectRank changes the role of the personalization vector from ranking to search. The topics are replaced with query entities, and the schema is weighted to form an ER graph in ObjectRank and its variations [28–31].

Similar to ObjectRank, the retrieved entities are used as the query vector to populate the personalized vector for an extended search. However, instead of using a monotonous probability transition matrix based on a predefined weighted graph, PERank populates the transition matrix based on the preference. For this purpose, PERank disassembles a whole graph into unit graphs that are created with classes or predicates to allow the user to obtain a customized ranking based on the preference. PERank exploits abundant links, especially in biomedical data, to retrieve related entities that cannot be obtained with the traditional text-based advanced entity search. To the best knowledge of the authors, PERank is the first method that improves text-based entity search by supporting the inter-topic search.

3. Inter-topic search based on PERank

Given a collection of linked datasets, an entity search returns the entities related to a keyword query and a search preference. The linked data consists of a set of entities $E = \{e_1, e_2, \dots, e_m\}$ and a set of links $L = \{l_1, l_2, \dots, l_n\}$ that connect the entities. It should be noted that since we are searching entities, only the instances in Assertion Box (A-Box), containing extensional knowledge and assertions about individuals (instances) [32], are considered as entities in this paper. For example, with the A-Box of DBpedia Ontology, it can be learned that “DBpedia:Upper_trunk” is an instance of “DBpediaOntology:Nerve”, and thus “DBpedia:Upper_trunk” is considered as an entity. The entities belong to a set of classes $C = \{c_1, c_2, \dots, c_d\}$ and the links belong to a set of predicates (properties) $P = \{p_1, p_2, \dots, p_f\}$. For example, Fig. 2(a) has six entities $\{e_1, e_3, \dots, e_6\}$ connected by ten links $\{l_1, l_2, \dots, l_{10}\}$. The entities belong to four classes in Fig. 2(b), such as the entity e_1 “Ethinamate” belonging to the class c_1 “Drugs”. The links belong to the ten predicates used to connect the classes in Fig. 2(b), such as the link l_2 belonging to the predicate p_1 “possibleDiseaseTarget”.

Text-based entity search indexes all the triples about an entity with different fields and returns the entity if the fields contain the query keywords [20]. For example, besides the triples for the entity “renal tubular acidosis” indexed in a mainly searched filed, a field named “class” with the value “disease” is also indexed for this entity. These fields are used as filters to restrict the search space related to the preference. In this same example, a search can be restricted in “disease” by searching the filed “class” with the preference “disease”. Instead of using these preferences as filters, PERank uses the links between the different entities that fit search preferences to tackle the inter-topic search scenarios, in which the topic related to the searched entities differs from the one related to the preference. For example, searching with the keywords “renal tubular acidosis” and the preference “drugs” to find the drugs to treat the disease “renal tubular acidosis” is considered as an inter-topic search.

3.1 Framework

Given a keyword query, PERank uses a set of entities retrieved from a keyword-based entity search engine to search the data with the help of the preference. Fig. 3 shows the graphical framework description of PERank, which consists of three parts: (1) query formulation, (2) search score formulation, and (3) Individual Personalized PageRank Vectors (IPPVs) computation and index, where the first two parts are on-line computations and the last part is off-line computation. During the on-line computation, a set of entity scores for each entity called IPPV is generated by the classes and predicates. These scores are used to form a ranking score in part 3 (i.e., search score formulation) with a query vector created in part 1 (i.e., query formulation) on-line.

3.2 Query formulation

In this phase, a query vector consisting of a pair of vectors, preference vector and query entity vector, used to feed PERank, are created.

A search preference is a set of terms $T = \{t_1, t_2, \dots, t_k\}$ that represents the search intention of a user. T is converted into a preference vector $\vec{p}$ in PERank. In practice, a preference can affect the authorities to flow to the entities in an RDF graph by two factors: class and predicate. Therefore, $\vec{p}$ is organized based on a class vector $\vec{cp}$ or a predicate vector $\vec{pp}$, where each entry is computed with the following equations if cp_i or pp_i contains t_j :

$$cp_i = \frac{1}{|T|} \sum_{t_j \in T} \frac{1}{|CP|} \quad (1)$$

$$pp_i = \frac{1}{|T|} \sum_{t_j \in T} \frac{1}{|PP|} \quad (2)$$

, where CP or PP is a set of classes or predicates containing t_j . For example in Fig. 2, given a preference $T = \{t_1: \text{“drug”}\}$, one class “Drugs” and two predicates “possibleDrug” and

“drugReference” contain the preference. Therefore, we can obtain $\vec{cp} = [1, 0, 0, 0]$ and $\vec{pp} = [0, 0.5, 0, 0, 0, 0, 0.5, 0, 0]$.

A query entity vector is an entity vector $\vec{qe} = [qe_1, qe_2, \dots, qe_m]$, where $qe_i = \frac{1}{|QE|}$ if $qe_i \in QE$. QE is a set of entities returned from an entity search with the keywords. For example, searching the entities related to the keyword “acidosis” in the data graph (shown in Fig. 2 (a)), QE will be returned as a set of entities $\{e_2, e_3\}$. And we can obtain the query entity $\vec{qe} = [0, 0.5, 0.5, 0, 0, 0]$.

3.3 Search score formulation

During the on-line computation, the query entity vector and preference vector are used as a pair of input to search IPPV scores and form a ranking score for all the entities in Algorithm 1. The query entities are obtained from the search engine using the keyword query (line 1) and then converted into the query entity vector (line 2). The preference is converted into a class vector ($\vec{cp}$) or a predicate vector $\vec{pp}$ (lines 3–4). Then, PERank uses $\vec{cp}$ or $\vec{pp}$ to get ranking scores from IPPVs (lines 4–12), and returns the sorted entities with the scores. The IPPVs are computed offline and will be introduced in the next section. Please note that, different from the top to bottom execution order of Algorithm 1, we introduce how the IPPVs are computed and used to form the PERank score (bottom to top).

Algorithm 1

PERank query execution algorithm.

Input: Preference T , Keywords Q

Output: Sorted Entities

1. Query entities $QE \leftarrow \text{entity Search}(Q)$
2. Query entity vector $\vec{qe} \leftarrow \text{query Entity Vector Convertor}(QE)$
3. Preference vector $\vec{cp} \leftarrow \text{preference Vector Convertor}(T)$
4. Preference vector $\vec{pp} \leftarrow \text{preference Vector Convertor}(T)$
5. Initial: Entity with ranking scores $PERank(CP, Q) \leftarrow 0, PERank(PP, Q) \leftarrow 0$,
6. **For** each entity $e \in QE$ **do**
7. Initial: $PPV(\vec{cp}, e) \leftarrow 0, PPV(\vec{pp}, e) \leftarrow 0$
8. **For** each entity $c_i \in CP$ **do**
9. $PPV(\vec{cp}, e) += cp_i \times IPPV(c_i, e)$
10. **For** each predicate $p_i \in PP$ **do**
11. $PPV(\vec{pp}, e) += pp_i \times IPPV(p_i, e)$
12. $PERank(CP, Q) += PPV(\vec{cp}, e)$

13. $PERank(PP, Q) += PPV(\vec{p}, e)$
14. Sort $PERank(CP, Q)$ and $PERank(PP, Q)$
15. Return $PERank(CP, Q)$ and $PERank(PP, Q)$

4. Computing PERank ranking scores based on IPPVs

In order to compute a ranking score with a pair of input, preference vector $\vec{p}$ and query entity vector $\vec{q}_e$, PERank adapts PPR, which populates a transition matrix with the preference vector and populates a distribution vector with the query entity vector. In order to handle the complexity of RDF graphs, a PERank score is assembled with Individual Personalized PageRank Vector scores (IPPVs) online. Fig. 4 shows an example of the computation of IPPVs that are demonstrated with a matrix, where each rows is a unit transition matrix populated by tracing a class in Fig. 2(b), and a column is an entity in Fig. 2(a). Therefore, for each pair of class and entity, an IPPV is computed and indexed. When a search job is initiated, the IPPVs in the gray boxes will be used to form a PERank ranking score for a query vector, which is generated indirectly from the keyword “acidosis” and directly from the preference “drug” and “diseases”.

An IPPV score is a PPR score calculated with an m dimensional unit query vector $\vec{q}_e$ where only the i th element of $\vec{q}_e$ is one and a unit preference vector $\vec{p}_i$ where only the j th element of $\vec{p}_i$ is one.

Definition 1. Individual Personalized PageRank Vector (IPPV)—For each pair of unit query vector $\vec{q}_e$ and unit preference vector $\vec{p}_i$, an $IPPV(\vec{q}_e, \vec{p}_i)$ is computed as:

$$IPPV(\vec{q}_e, \vec{p}_i)^{k+1} = d \times M_{\vec{p}_i} \times IPPV(\vec{q}_e, \vec{p}_i)^k + (1-d) \times \vec{q}_e \quad (3)$$

, where d is the damping factor. $IPPV(.,.)^{k+1}$ is the vector of scores at $(k+1)$ th iteration. $M_{\vec{p}_i}$ is the $m \times m$ dimensional individual transition matrix populated with $\vec{p}_i$

If none of u 's outbound nodes belong to c_j or none of u 's outbound links belong to p_j , an entry $m_{\vec{p}_i}$ of $M_{\vec{p}_i}$ is computed as:

$$m_{\vec{p}_i}(u, v) = \frac{1}{\#outLink(u)} \quad (4)$$

, otherwise:

$$m_{\vec{p}_i}(u, v) = \begin{cases} \frac{1}{\#outLink(u, j)} & v \text{ belongs to } c_j \\ \frac{1}{\#outLink(u, j)} & l_{u,v} \text{ belongs to } p_j \\ 0 & \text{else} \end{cases} \quad (5)$$

With a set of IPPVs, a Personalized PageRank Vector (PPV) can be formed.

Definition 2. Personalized PageRank Vector (PPV)—For a unit query vector $\vec{q}_i$ and a preference vector $\vec{p}_j$, where $\vec{p} = w_1 \vec{p}_1 + \dots + w_j \vec{p}_j$ and w_j is the weight of a unit preference vector $\vec{p}_j$, a $PPV(\vec{q}_i, \vec{p})$ is computed as:

$$PPV(\vec{q}_i, \vec{p}) = \sum w_j \times IPPV(\vec{q}_i, \vec{p}_j) \quad (6)$$

Based on the Linear Combination Theorem [33], a PERank score can be formed with PPVs:

$$PERank(\vec{q}, \vec{p}) = \sum w_i \times PPV(\vec{q}_i, \vec{p}) \quad (7)$$

, where $\vec{q} = w_1 \vec{q}_1 + \dots + w_i \vec{q}_i$, and w_i is the weight of a unit query entity vector $\vec{q}_i$. The weights w_j and w_i are the correspondence values of the query vector $\vec{q}$ and the preference vector in $\vec{p}$.

Given a linked dataset with m entities, n links, d classes and f predicates, the fully PPR computes every possible combination of personalization vectors and needs 2^m times of a standard PPV computation. With the Linear Combination Theorem, the cost of the fully PPR decreases into m times of a standard PPV computation. PERank simultaneously calculates a score with a query vector and a preference vector based on the linear combination with PPVs, each of which is linearly combined with IPPVs. IPPV calculates a score vector with a unit query vector and a unit preference vector, and the cost of an IPPV equals with the cost of a standard PPV computation. An individual PPV in PERank covers all the unit preference vectors (unit class preference vectors and unit predicate preference vectors) and costs $d + f$ times of a standard PPV computation. Therefore, covering all the combinations of the preference vectors and the query vectors costs $m(d + f)$ times of a standard PPV computation (or $(d + f)$ times of a standard PPR computation) in PERank.

5. Experiment

To evaluate PERank, we implemented PERank with JAVA_1.8 as shown in Fig. 5 based on a representative keyword-based entity search engine, Siren; we consider Siren as the baseline method. In the implemented system, keywords are entered in the search box. The topic and link boxes are used to catch the preference on classes or predicates. The results will be demonstrated literally and graphically. As the graphical representation for a search in Fig. 5 shows, searching the keyword “DB01259” (center node) obtains five entities (on the second circle) with the preference “drug” (on the first circle).

Our experiment is divided into three parts: (1) evaluation of the performance of IPPV computation, (2) comparison of the search results (effectiveness) of PERank with the different search strategies of Siren, (3) comparison of the effectiveness of PERank with the

classic PPR algorithms (TSPR [27] and ObjectRank [8]). All results can be found at (<https://github.com/zongnansu1982/PERankComputation/tree/master/results>).

5.1 Data and query sets

Ten datasets in Linked Life Data (<http://linkedlifedata.com/>) were used to form a biomedical graph, which included DailyMed (<http://dailymed.nlm.nih.gov/dailymed/>), Disesome (<http://disesome.eu/>), Disease Ontology (<http://disease-ontology.org/>), DrugBank (<http://www.drugbank.ca/>), Gene Ontology (<http://www.geneontology.org/>), HumanCyc (<http://humancyc.org/>), Gene-Disease Network (LHGDN) (<http://www.dbs.ifi.lmu.de/~bundschu/LHGDN.html>), REACTOME (<http://www.reactome.org/ReactomeGWT/entrypoint.html>), SIDER (<http://sideeffects.embl.de/>), and Symptom (http://symptomontologywiki.igs.umaryland.edu/wiki/index.php/Main_Page). The datasets have approximately 60 classes and 110 predicates, comprising a total of 400,000 entities and 1,357,000 links.

A pair of keyword query and preference was utilized as an input query in each search task. We randomly selected 20 names of diseases and drugs, and considered them as the keyword queries. The two topics, diseases and drugs, were used as the preference of search. All the combinations of the keyword queries and preferences pairs were generated for the intra- and inter-topic search tasks (shown in Appendix A).

5.2 Criteria for evaluation

To evaluate the quality of results from the keyword based entity search, we used precision that counted the number of returned entities both related with keyword and preference [34]. For those returned entities that were related to the keyword but not related to the preference, we used the validness to measure them.

Given keywords q and a preference p , the precision of the top K returns ($P@k$) was defined as:

$$P(q, p)_k = \frac{\# \text{ returned entities both related to } q \text{ and } p}{k} \quad (8)$$

Given keywords q and a preference p , the validness of the top K returns ($V@k$) was defined as:

$$V(q, p)_k = \frac{\# \text{ returned entities related to } q \text{ but not to } p}{k} \quad (9)$$

It should be noted that k was set to be a fixed number in $P@k$ and $V@k$. A search result with high precision (indicating high percentage of returned entities being related to both keyword query and preference) was considered as a good result. Compared to precision, high validness (indicating high percentage of entities only relating to the keyword) was still considered as an acceptable result, since, if the entities were keyword related, the true

related results could be found via tracing the links when the user browses the information of the entity. We have invited two medical Ph.D. students to evaluate the search results. They made correct annotations with 100% percent confidence with their knowledge or with the assistance of the third-party resources, such as PubMed (<https://www.ncbi.nlm.nih.gov/pubmed/>) and Up To Date (<https://www.uptodate.com/login>). Any uncertainty while annotating required the annotators to engage in discussion to reach consensus. The top 20 returned entities were manually checked for 40 searches of each task in the experiment.

5.3 Performance of IPPV computation

We computed the IPPVs for the 25,154 entities returned from Siren with the 20 keyword queries. The computation was conducted on a 2X Intel Xeon E5-2630 V4 processor (20 MB Cache, 2.2 GHZ) with a 128 GB memory running the Ubuntu 16.04 64-bit operating system. The IPPVs were separately computed for the 50 classes and 94 predicates that could be used as the preferences in the schema with Finger Prints [33]. This ranged from 100, 200, 400, 500, 600, 800 and 1000 steps respectively, and the performance is shown in Fig. 6. The time of IPPV computations for the returned entities on each class or predicate fluctuated a little, which proved that the whole computation time linearly increased along with the growing number of properties and classes. The time for each IPPV was also related to the number of steps of the random walk performed, and also linearly increased along with the growing of the steps.

5.4 Comparing PERank with Siren entity search strategies

In order to evaluate the search results of PERank, we designed three search strategies of Siren in the comparison: (1) query expansion that combined keywords with preference [35], (2) preference used as class filter, (3) preference used as predicate filter. Corresponding with (2) and (3), we used two kinds of PERank: (1) PERank Class and PERank Predicate.

The $P@k$ and $V@k$ were measured, where $k = 5, 10, 20$. A sample of the intra- and inter-topic search in top 5 that returns of the keyword “ethinamate” with the preferences “drug” and “disease” is shown in Appendix B, and the evaluation results are presented in Fig. 7. Please note that the value “NULL” in the appendix means empty. As the figure shows, searching with the preferences in general improves the inter-topic search, and only the query expansion and PERank improve the intra-topic search. Under the same circumstance, the class and predicate filters cannot improve the intra-topic search, but rather hinder the inter-topic search since they search the keyword related entities specific to those classes or predicates related to the preference (refer to some returns that are null in appendix B). This indicates that the preferences should perhaps only be used as the filters for intra-topic search [36]. PERank performs better than the other preference processing strategies in two kinds of search. PERank significantly increases the precision from 3.5% to 17.75% for PERank Class, 17.5% for PERank Predicate, validness from 4% to 28.85% for PERank Class and 25.51% for PERank Predicate in the inter-topic search. PERank, while improving the precision of the baseline from 41.5% to 50.5% for PERank Class and 65.25% for PERank Predicate, performs almost the same as the query expansion method that improves the precision of the baseline from 41.5% to 54.75% for the intra-topic search.

5.5 Comparing PERank with PPR algorithm

In this task, PERank class and predicate were compared with two PPR variations, TSPR and ObjectRank. In TSPR computation, an m -dimensional PPR vector $\vec{p}_i$ for each class was used to bias TSPR scores for the entities belonging to the class. In $\vec{p}_i$, each entry $p_j = \frac{1}{|c_i|}$ if p_j was an instance of c_i , otherwise $p_j = 0$, where $|c_i|$ was the cardinality of the instance set of c_i . Given a pair of input, keyword query and preference, the entities that were retrieved by Siren were sorted by the TSPR score based on the preference. In ObjectRank computation, a unit score vector was computed with an m dimensional unit PPR vector $\vec{q}_i$ and the probability transition matrix, where 80% of the weight was assigned to the predicates contained the preferences and the remaining 20% was equally distributed to other predicates. Given the input query, an ObjectRank score vector was dynamically generated by the linear combination of the unit score vectors for the entities retrieved by Siren.

In Appendices C and D, we show two samples of the search results in this task. In general, PERank performs better than both TSPR and ObjectRank in the intra and inter-topic searches.

As Fig. 8 shows, compared with TSPR, PERank Class improves precision and validness dramatically: from 14.75% to 50.5% on precision and 55.25% to 76.22% on validness in the intra-topic search; from 7.75% to 17.75% on precision and 14.5% to 28.85% on validness in the inter-topic search. The ranking strategy used in TSPR is less effective than PERank, because the results cannot be improved if all the entities returned from Siren are irrelevant. Different with TSPR, PERank uses returned entities to further search the related entities with links, which reduces the error entities from Siren.

Fig. 8 also shows that PERank Predicate outperforms ObjectRank: from 53.25% to 65.25% on precision and 56.46% to 78.10% on validness in the intra-topic search; from 2.25% to 17.5% on precision and 4.88% to 25.51% on validness in the inter-topic search. PERank customizes the weights of links corresponding to the preference, which flexibly teleports authorities toward the entities related to the preference. ObjectRank predefines the weights in the schema graph, which has to divvy a small portion of the authorities to preference-irrelevant entities.

6. Discussion

This study proposed a method to target inter-topic search for keyword-based entity search based on heterogeneous relationships in Linked Data. Even though there exists benchmarks and gold standards for testing entity search, such as Billion Triple Challenge (BTC) (<http://challenge.semanticweb.org/>), these benchmarks cannot be used in our evaluation since they only consider the relevancy between the query and entities. The preference, user search intentions, is not considered in these entity search challenges. Therefore, we conduct our experiments based on ten biomedical linked datasets and evaluated intra- and inter-topic search with two criteria: precision and validness at top k, based on the manually annotated entities. Since the data space in the experiment is limited, compared to the current size of biomedical Linked Data, the evaluation is not comprehensive from two aspects, (1) queries

are randomly generated, instead of using practical search cases, (2) criteria, such as Recall and F-measure, cannot be applied to provide tangible results. Therefore, building a new benchmark and gold standard for advanced biomedical entity search, where a keyword query and a preference are used as a pair of input, is considered as our future work.

PERank costs $m(d + f)$ times of a standard PPV computation to cover all the combinations of the preference vectors and the query vectors. There are scaling PPV computation methods, such as DrunkardMob [37], that can dramatically decrease the computation time. These methods can be applied for PERank to improve the performance. Another direction of our future work is to conduct PERank computations on-the-fly. With the entities retrieved from an entity search platform, a local network can be constructed based on these entities to train PERank. Therefore, without relying on the pre-indexed datasets, this variant can be used as a plugin for entity search in practice.

7. Conclusions

Search preferences are used as filters in the advanced keyword-based entity search that ignores relations in Linked Data. The preferences are used to restrict search space, which produces poor results for the inter-topic search. In this paper, we have introduced PERank that enhanced entity search by utilizing the relations resident in Linked Data. PERank draws upon an existing keyword-based entity search engine, and uses the returned entities and preferences to improve the search. We tested PERank with a set of linked life data and have compared it with the Siren-based search strategies and the existing PPR methods in light of effectiveness. The experiment has demonstrated that PERank returns more promising ranked entities in advanced search.

Acknowledgments

This work was funded in part by the Industrial Strategic Technology Development Program (10044494, WiseKB: Big data based self-evolving knowledge base and reasoning platform) supported by the Ministry of Science, ICT & Future Planning (MSIP, Korea). Partial funding also supported our work by Grant U24AI117966 to the University of California San Diego from the National Institutes of Health. We appreciate Dr. Hyeoneui Kim and Victoria Ngo for their suggestions on improving the manuscript.

References

1. Bizer C, Heath T, Berners-Lee T. Linked data-the story so far. *International journal on semantic web and information systems*. 2009; 5:1–22.
2. Blanco, R., Mika, P., Vigna, S. *The Semantic Web–ISWC 2011*. Springer; 2011. Effective and efficient entity search in RDF data; p. 83-97.
3. Bron, M., Balog, K., de Rijke, M. *Advances in Information Retrieval*. Springer; 2013. Example based entity search in the web of data; p. 392-403.
4. Liu TY. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*. 2009; 3:225–331.
5. Jindal V, Bawa S, Batra S. A review of ranking approaches for semantic search on Web. *Information Processing & Management*. 2014; 50:416–425.
6. Delbru R, Campinas S, Tummarello G. Searching web data: An entity retrieval and high-performance indexing model. *Web Semantics: Science, Services and Agents on the World Wide Web*. 2012; 10:33–58.

7. Wei W, Barnaghi P, Bargiela A. Rational Research model for ranking semantic entities. *Inf Sci.* 2011; 181:2823–2840.
8. Balmin, A., Hristidis, V., Papakonstantinou, Y. Objectrank: Authority-based keyword search in databases. *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30; VLDB Endowment; 2004.* p. 564-575.
9. Hogan A, Harth A, Umbrich J, Kinsella S, Polleres A, Decker S. Searching and browsing linked data with swse: The semantic web search engine. *Web semantics: science, services and agents on the world wide web.* 2011; 9:365–401.
10. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, RS., Peng, Y., Reddivari, P., Doshi, V., Sachs, J. Swoogle: a search and metadata engine for the semantic web. *Proceedings of the thirteenth ACM international conference on Information and knowledge management; ACM; 2004.* p. 652-659.
11. Navigli, R., Velardi, P. An analysis of ontology-based query expansion strategies. *Proceedings of the 14th European Conference on Machine Learning, Workshop on Adaptive Text Extraction and Mining; Cavtat-Dubrovnik, Croatia, Citeseer. 2003.* p. 42-49.
12. Halpin H, Lavrenko V. Relevance feedback between hypertext and Semantic Web search: Frameworks and evaluation. *Web Semantics: Science, Services and Agents on the World Wide Web.* 2011; 9:474–489.
13. Natsev, AP, Haubold, A., Teši , J., Xie, L., Yan, R. Semantic concept-based query expansion and re-ranking for multimedia retrieval. *Proceedings of the 15th ACM international conference on Multimedia; ACM; 2007.* p. 991-1000.
14. Efthimiadis, EN. A user-centred evaluation of ranking algorithms for interactive query expansion. *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval; ACM; 1993.* p. 146-159.
15. Jupp S, Malone J, Bolleman J, Brandizi M, Davies M, Garcia L, Gaulton A, Gehant S, Laibe C, Redaschi N. The EBI RDF platform: linked open data for the life sciences. *Bioinformatics.* 2014; 30:1338–1339. [PubMed: 24413672]
16. Consortium U. Activities at the universal protein resource (UniProt). *Nucleic acids research.* 2014; 42:D191–D198. [PubMed: 24253303]
17. Anguita A, García-Remesal M, de la Iglesia D, Maojo V. NCBI2RDF: enabling full RDF-based access to NCBI databases. *BioMed research international.* 2013; 2013:9.
18. Saleem M, Kamdar MR, Iqbal A, Sampath S, Deus HF, Ngomo ACN. Big linked cancer data: Integrating linked tcga and pubmed. *Web Semantics: Science, Services and Agents on the World Wide Web.* 2014; 27:34–41.
19. Castro LJG, McLaughlin C, Garcia A. Biotea: RDFizing PubMed Central in support for the paper as an interface to the Web of Data. *Journal of biomedical semantics.* 2013; 4:S5. [PubMed: 23734622]
20. Zong N, Lee S, Kim HG. Discovering expansion entities for keyword-based entity search in linked data. *Journal of Information Science.* 2015; 41:209–227.
21. Pérez-Agüera, JR., Arroyo, J., Greenberg, J., Iglesias, JP., Fresno, V. Using BM25F for semantic search. *Proceedings of the 3rd international semantic search workshop; ACM; 2010.* p. 2
22. Manning, CD., Raghavan, P., Schütze, H. *Introduction to information retrieval.* Cambridge university press; Cambridge: 2008.
23. Cheng, G., Ge, W., Qu, Y. Falcons: searching and browsing entities on the semantic web. *Proceedings of the 17th international conference on World Wide Web; ACM; 2008.* p. 1101-1102.
24. Belleau F, Nolin MA, Tourigny N, Rigault P, Morissette J. Bio2RDF: towards a mashup to build bioinformatics knowledge systems. *Journal of biomedical informatics.* 2008; 41:706–716. [PubMed: 18472304]
25. Page, L., Brin, S., Motwani, R., Winograd, T. *The PageRank citation ranking: Bringing order to the web.* Stanford InfoLab; 1999.
26. Haveliwala TH. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering.* 2003; 15:784–796.
27. Kamvar, S., Haveliwala, T., Manning, C., Golub, G. *Stanford University Technical Report.* Stanford: 2003. Exploiting the block structure of the web for computing pagerank.

28. Hwang H, Balmin A, Reinwald B, Nijkamp E. Binrank: Scaling dynamic authority-based search using materialized subgraphs. *IEEE Transactions on Knowledge and Data Engineering*. 2010; 22:1176–1190.
29. Chakrabarti, S. Dynamic personalized pagerank in entity-relation graphs. *Proceedings of the 16th international conference on World Wide Web*; ACM; 2007. p. 571-580.
30. Nie, Z., Zhang, Y., Wen, J-R., Ma, W-Y. Object-level ranking: bringing order to web objects. *Proceedings of the 14th international conference on World Wide Web*; ACM; 2005. p. 567-574.
31. Hristidis, V., Raschid, L., Wu, Y. WebDB. 2011. Scalable Link-based Personalization for Ranking in Entity-Relationship Graphs.
32. Zong N, Nam S, Eom JH, Ahn J, Joe H, Kim HG. Aligning ontologies with subsumption and equivalence relations in Linked Data. *Knowledge-Based Systems*. 2015; 76:30–41.
33. Fogaras D, Rácz B, Csalogány K, Sarlós T. Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments. *Internet Mathematics*. 2005; 2:333–358.
34. Blanco, R., Halpin, H., Herzig, DM., Mika, P., Pound, J., Thompson, HS., Duc, TT. Entity search evaluation over structured web data. *Proceedings of the 1st international workshop on entity-oriented search workshop (SIGIR 2011)*; New York: ACM; 2011.
35. Xu, S., Bao, S., Fei, B., Su, Z., Yu, Y. Exploring folksonomy for personalized search. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*; ACM; 2008. p. 155-162.
36. Dou, Z., Song, R., Wen, J-R. A large-scale evaluation and analysis of personalized search strategies. *Proceedings of the 16th international conference on World Wide Web*; ACM; 2007. p. 581-590.
37. Kyrola, A. Drunkardmob: billions of random walks on just a pc. *Proceedings of the 7th ACM conference on Recommender systems*; ACM; 2013. p. 257-264.

Appendix

A. Queries

Table A.1

Keywords and preferences pairs.

Intra-topic search		Inter-topic search	
Keywords	Preference	Keywords	Preference
Ethinamate	Drug	Ethinamate	Disease
Phenylpropanolamine	Drug	Phenylpropanolamine	Disease
Memantine	Drug	Memantine	Disease
Aliskiren	Drug	Aliskiren	Disease
Methdilazine	Drug	Methdilazine	Disease
Trovaflaxacin	Drug	Trovaflaxacin	Disease
Altretamine	Drug	Altretamine	Disease
Natamycin	Drug	Natamycin	Disease
Dexbrompheniramine	Drug	Dexbrompheniramine	Disease
Diatrizoate	Drug	Diatrizoate	Disease
Coats disease	Disease	Coats disease	Drug
Giant axonal neuropathy	Disease	Giant axonal neuropathy	Drug
Cerebrovascular disease	Disease	Cerebrovascular disease	Drug
Second degree AV block	Disease	Second degree AV block	Drug
Polydactyly	Disease	Polydactyly	Drug

Intra-topic search		Inter-topic search	
Keywords	Preference	Keywords	Preference
Hay-Wells syndrome	Disease	Hay-Wells syndrome	Drug
Paroxysmal nocturnal hemoglobinuria	Disease	Paroxysmal nocturnal hemoglobinuria	Drug
Xeroderma pigmentosum	Disease	Xeroderma pigmentosum	Drug
Dermatofibrosarcoma protuberans	Disease	Dermatofibrosarcoma protuberans	Drug
Non-Hodgkin lymphoma	Disease	Non-Hodgkin lymphoma	Drug

B. A sample of search results of using Siren baseline, Siren Query Expansion, Siren Class and Predicate Filter, and PERank

Table B.1
The top 5 returns of intra-topic search with the keyword “ethinamate” and the preference “drug”.

Baseline				Siren Query Expansion			
Entity	Name	Relevant to keywords	Fit Preference	Entity	Name	Relevant to keywords	Fit Preference
1	Drugbank:drugs/DB01031	Ethinamate	TRUE	1	Drugbank:drugs/DB01031	Ethinamate	TRUE
2	Drugbank:drugs/DB00609	Ethinamide	TRUE	2	Drugbank:drugs/DB00609	Ethinamide	TRUE
3	NULL	NULL	NULL	3	NULL	NULL	NULL
4	NULL	NULL	NULL	4	NULL	NULL	NULL
5	NULL	NULL	NULL	5	NULL	NULL	NULL
Siren Class Filter				Siren Predicate Filter			
1	Drugbank:drugs/DB01031	Ethinamate	TRUE	1	Drugbank:drugs/DB01031	Ethinamate	TRUE
2	Drugbank:drugs/DB00609	Ethinamide	TRUE	2	Drugbank:drugs/DB00609	Ethinamide	TRUE
3	NULL	NULL	NULL	3	NULL	NULL	NULL
4	NULL	NULL	NULL	4	NULL	NULL	NULL
5	NULL	NULL	NULL	5	NULL	NULL	NULL
PERank Class				PERank Predicate			
1	Dailymed:drugs/75	Trecator (Tablet, Film Coated)	TRUE	1	Bio2rdf:kegg/D00703	Ethinamate	TRUE
2	Drugbank:drugs/DB00609	Ethinamide	TRUE	2	Bio2rdf:kegg/D00591	Ethinamide	TRUE
3	Drugbank:drugs/DB01031	Ethinamate	TRUE	3	Drugbank: drugcategory/hypnoticsAndSedatives	Hypnotics and Sedatives	TRUE
4	Sider:drugs: 2761171	Ethinamide TRECRATOR	TRUE	4	Dailymed:drugs/75	Trecator (Tablet, Film Coated)	TRUE
5	Dailymed:drugs:1810	Diamox Sequels	TRUE	5	Drugbank:drugs/DB01031	Ethinamate	TRUE

Table B.2

The top 5 returns of inter-topic search with the keyword “ethinamate” and the preference “disease”.

Baseline					Siren Query Expansion				
Entity	Name	Relevant to keywords	Fit Preference	Entity	Name	Relevant to keywords	Fit Preference	Fit Preference	
1	Drugbank:drugs/DB01031	Ethinamate	TRUE	1	Drugbank:drugs/DB01031	Ethinamate	TRUE	FALSE	
2	Drugbank:drugs/DB00609	Ethinamide	TRUE	2	Disease:diseases	diseases	TRUE	TRUE	
3	NULL	NULL	NULL	3	Lhgd:association/55653	Liver diseases	FALSE	TRUE	
4	NULL	NULL	NULL	4	Lhgd:association/5649	Autoimmune Diseases	FALSE	TRUE	
5	NULL	NULL	NULL	5	Lhgd:association/5650	Calcinosis	FALSE	TRUE	
Siren Class Filter					Siren Predicate Filter				
1	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	
2	NULL	NULL	NULL	2	NULL	NULL	NULL	NULL	
3	NULL	NULL	NULL	3	NULL	NULL	NULL	NULL	
4	NULL	NULL	NULL	4	NULL	NULL	NULL	NULL	
5	NULL	NULL	NULL	5	NULL	NULL	NULL	NULL	
PERank Class					PERank Predicate				
1	Drugbank:drugs/DB00609	Ethinamide	TRUE	1	Drugbank:drugs/DB00609	Ethinamide	TRUE	FALSE	
2	Disease: diseases/1278	Renal tubular acidosis	TRUE	2	Drugbank:drugs/DB01031	Ethinamate	TRUE	FALSE	
3	Disease: diseases/3725	Renal tubular acidosis-osteopetrosis syndrome	TRUE	3	Disease: diseases/3725	Renal tubular acidosis-osteopetrosis syndrome	TRUE	TRUE	
4	Disease: diseases/992	Retinitis pigmentosa	FALSE	4	Disease: diseases/1278	Renal tubular acidosis	TRUE	TRUE	
5	Disease: diseases/440	Generalized epilepsy	FALSE	5	Disease: diseases/3566	Persistent hyperinsulinemic hypoglycemia of infancy	FALSE	TRUE	

C. A sample of search results of PERank Class and the TSPR

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table C.1

The top 5 returns of intra-topic search with the keywords “cerebrovascular disease” and the preference “disease” .

PERank Class					TSPR				
Entity	Name	Relevant to keywords	Fit Preference	Entity	Name	Relevant to keywords	Fit Preference	Entity	Fit Preference
1	Diseasome: diseases/1206	Xeroderma pigmentosum	TRUE	1	drugbank:drugs/DB00157	NADH	TRUE	FALSE	FALSE
2	Diseasome: diseases/347	Dystonia	TRUE	2	drugbank:drugs/DB00039	Palifermin	TRUE	FALSE	FALSE
3	Linkin:association/2917	Creutzfeldt-Jakob Disease, Familial	FALSE	3	dailymed:drugs/4	Kepivance (Injection)	TRUE	FALSE	FALSE
4	Diseasontology: DOID/0050254	acanthoceph haliasis	TRUE	4	drugbank:drugs/DB00048	Collagenase	TRUE	FALSE	FALSE
5	Drugbank:possible DiseaseTarget	Possible Disease Target	TRUE	5	drugbank:drugs/DB00619	Imatinib	TRUE	FALSE	FALSE

Table C.2
The top 5 returns of inter-topic search with the keywords “cerebrovascular disease” and the preference “drug” .

PERank Class					TSPR				
Entity	Name	Relevant to keywords	Fit Preference	Entity	Name	Relevant to keywords	Fit Preference	Fit Preference	
1	Drugbank:drugs/DB01233 Metoclopramide	TRUE	TRUE	1	diseasome: diseases/3725 Renal tubular acidosis-osteopetrosis syndrome	TRUE	FALSE	FALSE	
2	Drugbank:drugs/DB00742 Mannitol	TRUE	TRUE	2	diseasome: diseases/74 Alzheimer disease	TRUE	FALSE	FALSE	
3	Drugbank:drugs/2520 verapamil hydrochloridel	FALSE	TRUE	3	diseasome: diseases/1436 Analbuminemia	TRUE	FALSE	FALSE	
4	Drugbank:drugs/DB00850 Perphenazine	TRUE	TRUE	4	diseasome: diseases/2198 Dysalbuminemic hyperthyroxinemia	TRUE	FALSE	FALSE	
5	Drugbank:drugs/4168 metoclopramide	FALSE	TRUE	5	diseasome: diseases/2811 Insulin resistance, susceptibility to	TRUE	FALSE	FALSE	

D. A sample of search results of PERank Predicate and the ObjectRank

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table D.1
The top 5 returns of intra-topic search with the keywords “second degree AV block” and the preference “disease”.

PERank Predicate					ObjectRank			
Entity	Name	Relevant to keywords	Fit Preference	Entity	Name	Relevant to keywords	Fit Preference	
1 Dailymed:drugs/1061	CARDIZEM (Tablet, Coated)	TRUE	FALSE	1 Sider:Sideeffects/C0085614	AV block first degree	FALSE	TRUE	
2 Dailymed:drugs/537	Cardizem LA (Tablet)	TRUE	FALSE	2 Diseaseontology	Disease	FALSE	TRUE	
3 Dailymed:drugs/2385	Verapamil HCl (Injection)	TRUE	FALSE	3 Sider:Sideeffects/C0004245	Atrioventricular block	FALSE	TRUE	
4 Sider:side_effects: C0085614	AV block first degree	TRUE	TRUE	4 Sider:Sideeffects/C0264906	Second degree AV block	FALSE	TRUE	
5 Dailymed:drugs/2663	Metoclopramide (Solution)	TRUE	FALSE	5 Dbpedia:Atrioventricular_block	Atrioventricular block	FALSE	TRUE	

Table D.2
The top 5 returns of inter-topic search with the keywords “second degree AV block” and the preference “drug” .

PERank Predicate					ObjectRank			
Entity	Name	Relevant to keywords	Fit Preference	Entity	Name	Relevant to keywords	Fit Preference	
1 Drugbank:drugtype/smallMolecule	Small Molecule	FALSE	TRUE	1 Sider:Sideeffects/C0085614	AV block first degree	FALSE	FALSE	
2 Drugbank:drugs/DB01233	Metoclopramide	TRUE	TRUE	2 Diseaseontology	Disease	FALSE	FALSE	
3 Drugbank:drugtype/approved	Approved drug	FALSE	TRUE	3 Sider:Sideeffects/C0004245	Atrioventricular block	FALSE	FALSE	
4 Sider:side_effects: C0085614	AV block first degree	TRUE	FALSE	4 Sider:Sideeffects/C0264906	Second degree AV block	FALSE	FALSE	
5 Disease: diseases/443	Giant axonal neuropathy	FALSE	FALSE	5 Dbpedia:Atrioventricular_block	Atrioventricular block	FALSE	FALSE	

- We developed a keyword-based entity search algorithm, PERank, in tandem with existing entity search methods for inter-topic biomedical entity search.
- PERank adapts Personalized PageRank (PPR) to use heterogeneous relations in linked data to search entities across topics.
- With ten Linked Life Data sets and two query sets (intra-topic and inter-topic search), PERank outperformed the baseline search and existing PPR methods.

Fig. 1.
An example of failure in Linked Life Data when it meets an inter-topic entity search.

Fig. 2.
An example of an RDF graph in Biomedical domain.

Fig. 3.
The framework of PERank.

Fig. 4.
The computation of PERank based on IPPVs with tracing classes of Fig. 2.

Fig. 5.
Screenshot of PERank.

(a) IPPV computation for classes

(b) IPPV computation for properties

Fig. 6.
Performance of IPPV computation for classes and properties.

Fig. 7.

Effectiveness of PERank comparing with other preference processing strategies

Fig. 8.
Effectiveness of PERank comparing with the TSPR and ObjectRank.