Graph Analytical Re-ranking for Entity Search

Takahiro Komamizu
Nagoya University
Nagoya, Japan
taka-coma@acm.org

Abstract

Entity search is a fundamental task in Linked Data (LD). The task is, given a keyword search query, to retrieve a set of entities in LD which are relevant to the query. The state-of-the-art approaches for entity search are based on information retrieval technologies such as TF-IDF vectorization and ranking models. This paper examines the approaches by applying a traditional evaluation metrics, recall@k, and shows ranking qualities still room left for improvements. In order to improve the ranking qualities, this paper explores possibilities of graph analytical methods. LD is regarded as a large graph, graph analytical approaches are therefore appropriate for this purpose. Since query-based graph analytical approaches fit to entity search tasks, this paper proposes a personalized PageRank-based re-ranking method, PPRSD (Personalized PageRank based Score Distribution), for retrieved results by the state-of-the-art. The experimental evaluation recognizes improvements but its results are not satisfactory, yet. For further improvements, this paper reports investigations about relationship between queries and entities in terms of path lengths on the graph, and discusses future directions for graph analytical approaches.

1 Introduction

Linked Data (LD) [BHB09] which started by Sir Tim Berners-Lee has become an important knowledge source, and entity search for LD [PMZ10] is a fundamental task which retrieves entities in LD for query keywords. Entity search is important for users who investigate entities themselves as well as relationships among entities. Due to its importance, several open tasks for entity search have been published (for instance, INEX-LD [WKC^{+}12], QALD [UNH^{+}17], and DBpedia-Entity [HNX^{+}17]). DBpedia-Entity is the most recent open entity search task composing the existing open entity search tasks and contains comprehensive evaluation results for existing entity search methods. Therefore, this paper deals with this task.

In DBpedia-Entity task, recent methods are inspired from information retrieval domains, such as BM25 and language model (see their website[1]), however, there are few methods using graph analytical methods (e.g., PageRank [PBMW99]). The existing methods are based on occurrences of terms; BM25 is a common ranking model-based TF-IDF vectorization, language model considers probabilities of co-occurrence of terms, and, fielded extension methods over the former basic methods are also included. Fielded extension methods give high weights for important attributes of documents (e.g., titles of Web pages). On the other hand, interestingly, there are few methods using graph analytical methods such as PageRank, even though LD is represented as a graph in nature. This raises a question: *Are graph analytical methods not appropriate for entity search tasks?*

To answer the question, this paper firstly analyzes the existing methods. [HNX^{+}17] indicates that existing methods achieve 0.46 NDCG@10 score and 0.55 NDCG@100 score, but it is not clear how far the achievements from goals are. NDCG (Normalized Discounted Cumulative Gain) is a standard way of ranking evaluation, which reasonably compares ranking methods, but it hides potentials for improvement. Therefore, this paper applies a traditional evaluation metrics, *recall@k*, which is a ratio of relevant answers in top-k results over the total number of relevant answers. Thus, recall@k indicates that how many rele-

[1] http://tiny.cc/dbpedia-entity

vant answers are absent in top-k results. The analysis results shown the Total column in Table 1, recall@10, recall@100, and recall@1000 are maximally 0.2872, 0.6912, and 0.8708, respectively.

The investigation results indicate that there are still room left for improving rankings. The low recall for top-10 results and the high recall for top-1000 results imply that large amount of relevant results are within 1000 results but most of them are below top-10. Therefore, this paper attempts to improve the ranking by reranking, which arrange the ranking by applying different ranking criteria. It is reasonable to take graph topological features into account due to the nature of LD. Therefore, this paper applies graph analytical methods for re-ranking. The result for the re-ranking method is expected to be an answer to the question.

In consequence, this paper arranges the aforementioned question to the following question. *Do graph analysis-based re-ranking methods improve the ranking quality?* This paper attempts to take graph analytical methods into account and proposes a re-ranking method *PPRSD* (Personalized PageRank based Score Distribution) which distributes calculated relevance scores by the state-of-the-art in a personalized PageRank manner. Test of PPRSD gives the following answer, the graph analysis-based re-ranking method can improve the ranking quality but the improvement is not very significant.

In order to find future directions based on graph analytical methods for improving entity search, this paper performs investigations and provides insights. This paper poses a question for results of the preliminary evaluation by recall@k, that is, *why recall@1000 is not perfect yet?* To answer the question, this paper investigates relationship between query terms and relevant entities for the query, and the investigation reveals that some terms only exist on distant literals from relevant entities. Additionally, this paper obtains a clue for selection of predicates connecting to literals w.r.t. different distances from the entities. Based on these investigations, this paper puts discussion on future directions based on graph analytical approaches for entity search.

The following sections discuss the detail for getting the answers to the questions. Section 2 introduces briefly the state-of-the-art shown in [HNX$^+$17] and showcases the preliminary evaluation in terms of recall@k metrics, and Section 3 explains the idea and detail of PPRSD, and Section 4 evaluates the state-of-the-art and PPRSD using the test collection and shows the answers to the aforementioned questions. Section 5 displays additional investigations and insights for the future directions, and Section 6 concludes this paper.

2 State of Current Entity Search

This work explores the future directions of entity search, to this end, this paper investigates the current state of entity search, especially this paper sticks to a leading benchmark, DBpedia-Entity v2 [HNX$^+$17].

As shown in the benchmark, there are various approaches which are mainly based on information retrieval and natural language processing techniques. The list of approaches include fundamental approaches: BM25 [RZ09], BM25-CA [RZ09], LM (Language Modeling) [PC98], SDM (Sequential Dependency Model) [MC05], PRMS (Probabilistic Model for Semistructured Data) [KXC09], and MLM-all (Mixture of Language Models) [OC03]; fielded extension approaches: MLM-CA [OC03], FSDM (Fielded Sequential Dependence Model) [ZKN15], and BM25F-CA [RZ09]; extended approaches by entity linking technique [HBB16] for query: LM-ELR [HBB16], SDM-ELR [HBB16], and FSDM-ELR [HBB16].

These works are based on a fielded document construction method in [Has18]. As an overall structure, each entity has 1000 fields together with three additional fields. The 1000 fields are corresponding with top 1000 frequent predicates in DBpedia, and the additional fields are heuristically constructed such that one is "name" field which is constitution of predicates `rdfs:label` and `foaf:name`; another is "types" field which contains `rdf:type` predicate and predicates ending in "subject", and the other is "contents" field which holds the contents of all fields of connected entities except those connected by `owl:sameAs` to remove same entities in different languages. Aforementioned approaches use parts of the fielded documents as follows: BM25, LM and SDM use the contents field; and MLM-all, PRMS and FSDM use top-10 fields. The field extension approaches are differentiated by settings of field weights (e.g., MLM-all uses equal weights for all fields, while PRMS learns weights for fields).

To investigate the qualities of these approaches, this paper tests more intuitive metrics recall@k in addition to NDCG which is shown in [HNX$^+$17]. The NDCG results are copied to Table 4 (rows of not *-ed method names correspond to the original results shown in [HNX$^+$17]). The NDCG result shows comparative ranking qualities among these approaches. While, NDCG is not a clear indicator for distances from goals. Therefore, this paper investigates more clear indicator, recall@k (Eqn. 1) which reveals ratio of relevant results in top-k.

$$recall@k = \frac{\text{the number of relevant items in top-k}}{\text{the total number of relevant items}} \tag{1}$$

Table 1 displays recall@k ($k \in \{10, 100, 1000\}$) and it indicates that more than 80% of relevant results are

Table 1: Recall@k (k = 10, 100, 1000). Each row corresponds with existing approaches, and the last row is maximum recall score among them. For each column, the best score is boldface, underlined, and lined in the bottom. The bottom row indicates gaps from recall@k values (k = 10, 100) from recall@1000, which claims that large amount of relevant results are below top-10.

Model	SemSearch ES			INEX-LD			ListSearch			QALD-2			Total		
	@10	@100	@1000	@10	@100	@1000	@10	@100	@1000	@10	@100	@1000	@10	@100	@1000
BM25	.2563	.6669	.9280	.1730	.4860	.7554	.1093	.4598	.7221	.1891	.4677	.6929	.1823	.5175	.7703
PRMS	.3719	.7499	.9412	.2312	.5339	.7796	.1839	.5476	.7525	.2273	.5428	.7420	.2522	.5919	.8009
MLM-all	.3887	.7705	.9412	.2343	.5527	.7796	.1840	.5655	.7525	.2280	.5706	.7420	.2571	.6136	.8009
LM	.3812	.8236	.9412	.2425	.5807	.7796	.1899	.5772	.7525	.2355	.5910	.7420	.2607	.6413	.8009
SDM	.3884	.8581	.9865	.2409	.6224	.8567	.1987	.6121	.8256	.2398	.5921	.7991	.2659	.6674	.8633
LM-ELR	.3863	.8278	.9412	.2364	.5894	.7796	.1913	.5940	.7536	.2474	.5909	.7401	.2646	.6483	.8006
SDM-ELR	.3898	.8581	**.9865**	.2366	.6307	.8567	.2105	.6180	.8256	.2589	.6172	.7991	.2739	.6782	.8633
MLM-CA	.4096	.7843	.9420	.2249	.5917	.8051	.1861	.5834	.8038	.2377	.5953	.7894	.2639	.6370	.8329
BM25-CA	.3991	.8326	.9766	.2372	.6266	**.8603**	**.2110**	**.6261**	**.8431**	**.2650**	.6157	**.8164**	.2782	.6727	**.8708**
FSDM	.4459	.8515	.9581	.2390	.6153	.8191	.1980	.5999	.8175	.2466	.6102	.7970	.2812	.6667	.8455
BM25F-CA	.4097	**.8707**	.9704	**.2607**	**.6526**	.8544	.2042	.6189	.8325	.2548	**.6341**	.8157	.2811	**.6912**	.8653
FSDM-ELR	**.4536**	.8539	.9562	.2477	.6253	.8191	.2022	.6075	.8162	.2507	.6275	.7970	**.2872**	.6765	.8450
max	.4536	.8707	.9865	.2607	.6526	.8603	.2110	.6261	.8431	.2650	.6341	.8164	.2872	.6912	.8708
gap	.5329	.1158	—	.5996	.3077	—	.6321	.2170	—	.5514	.1823	—	.5836	.1796	—

Table 2: Recall@k (k = 10, 100). Each row corresponds to the maximum recall@k value among re-ranked existing approaches.

Re-ranking method	SemSearch ES		INEX-LD		ListSearch		QALD-2		Total	
	@10	@100	@10	@100	@10	@100	@10	@100	@10	@100
PageRank	.1545	.4664	.1171	.3639	.1059	.4438	.1561	.4519	.1344	.4198
Personalized PageRank	.1632	.4779	.1228	.3822	.1146	.4524	.1613	.4587	.1397	.4355

included in top-1000 but only 20% to 45% of them are included in top-10, which indicates there are room left for improving rankings. The recalls are calculated on the top-1000 results presented in the benchmark data[2]. The boldface and underlined cells in the table show maximum recall scores for tasks and k. All methods have low recall@10 as well as recall@100, but still high recall@1000, meaning that ranking performance should be improved. The gap row in the table emphasizes that top-10 results have large room left for improvements.

3 PageRank-based Re-ranking

This work attempts to improve the ranking qualities by graph analytical re-ranking methods. LD is modeled as a labeled graph, it is therefore reasonable to apply graph analytical approaches to evaluate values of entities. In particular, this paper explores feasibility of PageRank [PBMW99], which is popular graph analytical methods to originally evaluate Web pages and has been applied for many other domains.

This paper models LD data as data graph (Def. 1)

Definition 1 (Data Graph) *Given LD data, data graph $G = (V, E)$ is a graph, where set $V = R \cup L \cup B$ of vertices are union of set R of entities, set L of literals and set B of blank nodes, and set $E \subseteq V \times P \times V$ of edges between vertices with predicates P as labels.* □

The subsequent sections introduce naïve baseline approaches and the proposed re-ranking method, PPRSD. Section 3.1 introduces re-ranking methods via PageRank [PBMW99] and personalized PageRank [Hav02], and introduces a preliminary evaluation of these methods. Then, Section 3.2 explains PPRSD which utilizes both results of the state-of-the-art and advantages of personalized PageRank.

3.1 Naïve Graph Analytical Re-ranking

As discussed above, graph analytical approaches are reasonable for re-ranking criteria, however, with a little consideration, global evaluation methods like PageRank do not make sense for ranking entities with respect to input keyword queries. Roughly speaking, PageRank evaluates vertices having lots of incoming links as important. Therefore, when PageRank is applied to the data graph G, PageRank gives an order of vertices which is independent from input queries. Examinations for the global rankings show bad results (this paper does not include this because it is obvious).

In order to test PageRank and personalized PageRank in a re-ranking manner, this work utilizes an insight from the recall@k results in Table 1. The insight is that the top-1000 results by existing methods include more than 80% of relevant results. Thus, the idea of re-ranking with PageRank and personalized PageRank is to filter top-1000 result entities by the existing methods and to apply the graph analytical approaches. To do so, an *induced subgraph* (Definition 2) for the top-1000 result entities are extracted.

[2] `https://github.com/iai-group/DBpedia-Entity/tree/master/runs/v2`

Definition 2 (Induced Subgraph) *Given set V' of vertices, induced subgraph $G' = (V', E')$ of graph $G = (V, E)$ over V' is a subgraph of G such that $V' \subseteq V$ and $E' = (V' \times V') \cap E$.* □

On the induced subgraph G' extracted from top-1000 results, PageRank and personalized PageRank values are calculated as Eqn. 2 and Eqn. 3. In Eqn. 2, $\mathbf{pr}$ is a PageRank vector with 1000 length, A is a 1000×1000 adjacency matrix of G', $\mathbf{e}$ is 1000-length vector which elements are all 1, and d is a damping factor which is the probability of random jumps. Similarly, in Eqn 3, $\mathbf{ppr}_q$ is a 1000-length PageRank vector for query q, A is an adjacency matrix as PageRank, $\mathbf{s}$ is 1000-length personalized vector for q, which elements corresponding with matching entities for q are 1 and other elements are 0, and d is a damping factor.

$$\mathbf{pr} = (1-d) \cdot \mathbf{pr}A + d \cdot \mathbf{e} \tag{2}$$

$$\mathbf{ppr}_q = (1-d) \cdot \mathbf{ppr}_q A + d \cdot \mathbf{s} \tag{3}$$

A preliminary experiment over these naïve re-ranking methods shows worse results than the state-of-the-art. The preliminary experiment tests the feasibility of aforementioned methods (PageRank and personalized PageRank-based re-ranking methods) on the DBpedia-Entity v2 benchmark [HNX+17]. The re-ranking approaches are applied for all the state-of-the-art methods listed in Table 1. Table 2 displays maximum recall@k values among the applied methods of PageRank and personalized PageRank, separately. Amongst PageRank and personalized PageRank, personalized PageRank has achieved better performance than PageRank, therefore, taking relevance to queries into account results better ranking qualities. Comparing recall@k of the state-of-the-art shown in Table 1, the re-ranking methods are mostly worse then them. Consequently, re-ranking methods should more rely on the state-of-the-art.

3.2 Re-ranking by Score Distribution

The preliminary evaluation on the naïve re-ranking methods reveal two facts: one is personalized PageRank-based re-ranking is superior to PageRank-based re-ranking, and the other is the state-of-the-art are still more powerful than simple graph analytical approaches. Therefore, the facts suggest that personalized PageRank-based method with utilizing results of the state-of-the-art can be a better choice. The rest of this section introduces how to realize it.

The main idea of the proposed approach is that utilizing relevance scores for re-ranking algorithm via personalized PageRank. The state-of-the-art rank entities by their own relevance scores, the scores indicate relative relevance degrees among the resulting entities. That is, there are more or less gaps on relevance scores than those on ranks. Additionally, the relevance scores are more sophisticated than just counting matching entities as naïve personalized PageRank-based re-ranking approach ($\mathbf{s}$ in Eqn. 3).

To realize this idea, this work arranges the personalized PageRank formulation shown in Eqn. 3 to include the relevance scores calculated by the state-of-the-art as Eqn. 4 called **PPRSD** (stands for **P**ersonalized **P**age**R**ank based **S**core **D**istribution).

$$\mathbf{pprsd}_q = (1-d) \cdot \mathbf{pprsd}_q A + d \cdot \mathbf{t} \tag{4}$$

where $\mathbf{pprsd}_q$ is a 1000-length relevance score vector of PPRSD. The personalized vector $\mathbf{s}$ is redefined as $\mathbf{t}$, where each element t_i of entity $v_i \in V'$ stores a relevance score of q to v_i calculated by one of the state-of-the-art method. Log likelihood-based relevance scores (i.e., LM, MLM, SDM, FSDM, PRMS, and their variations) are negative values in nature, therefore, these scores are converted to positive numbers by applying exponential function. In addition, the converted scores are quite small (e.g., 10^{-34}) because the values in the log function are products of probabilities, therefore, the converted scores are multiplied by positive number so as to make the scores comparable with those of the other methods. As PageRank-based methods compute the relevance score vectors (i.e., $\mathbf{pr}$ in Eqn. 2 and $\mathbf{ppr}$ in Eqn. 3), PPRSD also computes the relevance score vector, $\mathbf{pprsd}$, by the power method. Result entities ranked by PPRSD are of ordering in the relevance scores.

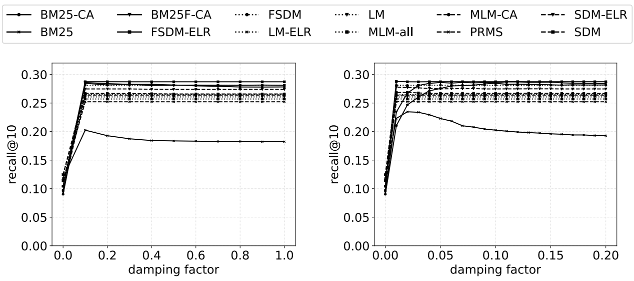
4 Experimental Evaluation

The experiment in this paper attempts to confirm the re-ranking method, PPRSD, improves the ranking qualities in terms of both recall@k and NDCG@k. Since PPRSD relies on the results of the state-of-the-art, this experiment uses the standard benchmark dataset [HNX+17][3] of entity search on DBpedia. Relevance scores for entities in the state-of-the-arts are obtained from the website of the benchmark[4]. This experimental evaluation attempts to answer the following question: *Does the re-ranking method improve the state-of-the-are? And, how large or small the improvements are?* In order to answer the question, PPRSD and the state-of-the-art are compared by recall@k (Eqn. 1) and NDCG@k (Eqn. 5) which is a ratio of DCG@k (Eqn. 6) over the ideal value of DCG@k referred as to IDCG@k.

$$NDCG@k = \frac{DCG@k}{IDCG@k} \tag{5}$$

[3] http://tiny.cc/dbpedia-entity

[4] https://github.com/iai-group/DBpedia-Entity/tree/master/runs/v2



(a) Damping factor in 0 to 1. (b) Damping factor in 0 to 0.2.

Figure 1: Effect of damping factor. Lines represent base methods for PPRSD. (a) shows recall@10 values for damping factor 0 to 1 and realizes damping factors in 0 to 0.2 are optimal, therefore, (b) shows that range in fine granularity.

$$DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{6}$$

The subsequent sections discuss the comparison of the ranking qualities between the original methods and the re-ranked methods by PPRSD. More specifically, Section 4.1 introduces an empirical study for determining damping factor d in Eqn. 4, and, based on the choice of the damping factor, Section 4.2 discusses the comparison between the PPRSD-based methods over the original methods.

4.1 Effect of Damping Factor

Figure 1 showcases effects of damping factor in various state-of-the-art which PPRSD is applied, and it reveals that smaller damping factor (i.e., around 0.1) achieves the best performances. In the figure, horizontal axis expresses damping factor which ranges 0 to 1.0 in Figure 1(a) and 0 to 0.2 in Figure 1(b), vertical axis represents recall@10, and lines in the figure represent the state-of-the-art. FSDM-ELR and BM25F-CA perform the best among the state-of-the-art in the figure around damping factor 0.1. In consequence, keeping 10% of relevance scores is the best choice for PPRSD, so $d = 0.1$ is used for the later experiments.

4.2 Overall Evaluation

Table 3 and Table 4 display the comparisons of values of recall@k for the former and NDCG@k for the latter among PPRSD and the state-of-the-art. The *Model* row of the table represents tasks of entity search and values of k, and the left-most column shows lists of the state-of-the-art and re-ranked versions (which are represented by *) of them by PPRSD. In addition, each group of rows corresponding with a method includes *imp.* row which emphasizes the improvement ratio by PPRSD. Cells contain recall@k values, and the best value in a row is emphasized by boldface and underline.

Table 3 shows PPRSD successfully improves ranking qualities of the state-of-the-art. The *Total* column represents the ranking qualities among all tasks. The column indicates that 7 over 12 methods have been improved by PPRSD in recall@10 and 8 methods have also been improved by PPRSD but 2 methods have been degraded the ranking qualities in recall@100. This indicates that PPRSD successfully improves the ranking qualities. Note that PPRSD improves not only elementary approaches (e.g., BM25) but also more sophisticated approaches (e.g., BM25F-CA and FSDM-ELR).

Table 4 also shows PPRSD successfully improves ranking qualities of the state-of-the-art. Recall@k and NDCG@k obviously have correlation, therefore, the improvements should be confirmed as the success shown in recall@k. As expected, the best results are all of PPRSD-based methods. It is worthy to note that the improvement ratios shown in *imp.* are larger than those of recall@k, indicating that PPRSD improves the rankings not only by just more relevant entities in the rankings but also better positions of relevant entities in the rankings. Since NDCG@k is good at relative comparison between rankings, the results confirm the improvements of rankings. For example, BM25-CA* is the best ranking method in terms of recall@10 for QALD-2 task, however, BM25F-CA* is superior to BM25-CA* and the best in terms of NDCG@10 for the same task. This means that BM25F-CA* have less relevant entities in top-10 rankings but more relevant entities are in the earlier positions in the top-10 rankings. Consequently, evaluation based on NDCG@k confirms the ranking improvements by PPRSD.

5 Investigation for Improvements

This section explores possibilities of further improvements for the state-of-the-art and PPRSD-based reranking methods in the following aspect: *Why recall@1000 is not 100% yet?* Since PPRSD is based on the state-of-the-art, the upper bound of ranking qualities is limited by them and improving the state-of-the-art is also important for further improvement of PPRSD. Therefore, this work investigates the reason why top-1000 results have not been perfect yet. To answer the question, this paper investigates path lengths from relevant entities to entities which literals contain an input keyword query term (detail in Section 5.1). The investigation reveals that there are still space left for including literals within larger distances (i.e., 3, 4, and 5 hops). Obviously, taking longer paths (or sequences of predicates) into account entails explosion of the number literals included into documents of entities. As a result, each entity gets noisy documents, and it is easy to imagine that the noisy documents degrade

Table 3: Recall@k (k=10, 100). Model indicates task types of queries, and top-k indicates the selected k values (10 or 100). Each cell contains a recall@k value for corresponding condition. For each column, the best score is boldface and underlined. The most-left column lists the state-of-the-art and re-ranked versions of them by PPRSD (corresponding with *-ed names). Each group of rows corresponding with the state-of-the-art includes imp. row indicating the ratio of the improvement by PPRSD.

Model	SemSearch ES		INEX-LD		ListSearch		QALD-2		Total	
	@10	@100	@10	@100	@10	@100	@10	@100	@10	@100
BM25	.2563	.6669	.1730	.4860	.1093	.4598	.1891	.4677	.1823	.5175
BM25*	.2735	.6952	.1867	.5144	.1279	.4809	.2036	.5044	.1983	.5466
imp.	+6.52%	+3.64%	+5.38%	+5.21%	+13.54%	+3.70%	+7.19%	+6.33%	+7.57%	+4.70%
PRMS	.3719	.7499	.2312	.5339	.1839	.5476	.2273	.5428	.2522	.5919
PRMS*	.3719	.7499	.2312	.5339	.1839	.5476	.2273	.5428	.2522	.5919
imp.	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MLM-all	.3887	.7705	.2343	.5527	.1840	.5655	.2280	.5706	.2571	.6136
MLM-all*	.3887	.7705	.2343	.5527	.1840	.5655	.2280	.5706	.2571	.6136
imp.	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
LM	.3812	.8236	.2425	.5807	.1899	.5772	.2355	.5910	.2607	.6413
LM*	.3812	.8222	.2425	.5807	.1899	.5772	.2355	.5910	.2607	.6410
imp.	0.00%	-0.14%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	-0.03%
SDM	.3884	.8581	.2409	.6224	.1987	.6121	.2398	.5921	.2659	.6674
SDM*	.3925	.8602	.2409	.6232	.1991	.6134	.2402	.5921	.2671	.6684
imp.	+1.11%	+0.24%	0.00%	+0.08%	+0.05%	+0.16%	+0.17%	0.00%	+0.41%	+0.13%
LM-ELR	.3863	.8278	.2364	.5894	.1913	.5940	.2474	.5909	.2646	.6483
LM-ELR*	.3863	.8231	.2364	.5894	.1913	.5945	.2474	.5909	.2646	.6473
imp.	0.00%	-0.43%	0.00%	0.00%	0.00%	+0.08%	0.00%	0.00%	0.00%	-0.12%
SDM-ELR	.3898	.8581	.2366	.6307	.2105	.6180	.2589	.6172	.2739	.6782
SDM-ELR*	.3936	.8590	.2366	.6305	.2107	.6190	.2589	.6172	.2749	.6786
imp.	+1.03%	+0.09%	0.00%	-0.03%	+0.10%	+0.18%	0.00%	0.00%	+0.37%	+0.06%
MLM-CA	.4096	.7843	.2249	.5917	.1861	.5834	.2377	.5953	.2639	.6370
MLM-CA*	.4096	.7843	.2249	.5919	.1861	.5834	.2377	.5953	.2639	.6371
imp.	0.00%	0.00%	0.00%	+0.03%	0.00%	0.00%	0.00%	0.00%	0.00%	+0.02%
BM25-CA	.3991	.8326	.2372	.6266	.2110	.6261	.2650	.6157	.2782	.6727
BM25-CA*	.4085	.8345	.2350	.6301	**.2151**	**.6278**	**.2701**	.6329	.2826	.6795
imp.	+2.26%	+0.12%	-0.38%	+0.35%	+2.27%	+0.38%	+0.57%	+2.79%	+1.33%	+0.97%
FSDM	.4459	.8515	.2390	.6153	.1980	.5999	.2466	.6102	.2812	.6667
FSDM*	.4463	.8528	.2390	.6156	.1980	.5998	.2466	.6103	.2813	.6671
imp.	+0.09%	+0.15%	0.00%	+0.05%	0.00%	-0.02%	0.00%	+0.02%	+0.04%	+0.06%
BM25F-CA	.4097	.8707	.2607	.6526	.2042	.6189	.2548	.6341	.2811	.6912
BM25F-CA*	.4218	**.8753**	**.2628**	**.6555**	.2047	.6226	.2613	**.6423**	.2865	**.6963**
imp.	+2.95%	+0.45%	+1.42%	+0.34%	+0.73%	+0.69%	+2.00%	+1.39%	+1.99%	+0.72%
FSDM-ELR	.4536	.8539	.2477	.6253	.2022	.6075	.2507	.6275	.2872	.6765
FSDM-ELR*	**.4540**	.8552	.2477	.6256	.2022	.6075	.2507	.6277	**.2873**	.6769
imp.	+0.09%	+0.15%	0.00%	+0.05%	0.00%	0.00%	0.00%	+0.03%	+0.03%	+0.06%

ranking qualities. To obtain hints for preferable paths for literals, this paper investigates the commonalities of tail predicates in the paths (detail in Section 5.2). The investigation reveals that tail predicates should be different for different lengths of the paths.

5.1 Distance from Query Term

The state-of-the-art rely on terms occurring within two hops at most, modeled as fielded documents. Section 2 introduces the fields of entities taken into account for the state-of-the-art, and the contents field includes contents of one-hop away entities. This implies that no method considers terms occurring within longer hops away.

The fielded document construction limits the possibilities to reach to the relevant results due to the absence of query terms in the documents. This fact is estimated from the preliminary evaluation on recall@k in Table 1, that is, recall@1000 values are less than 86% (except SemSearch ES task which is designed for direct matching with terms). In other words, 14% are below top-1000 results.

In order to answer question *why recall@1000 is not perfect?*, this paper attempts to realize the relation between relevant answers and the numbers of hops from query terms. To this end, this work investigates the minimum distances from relevant entities to query terms by performing SPARQL queries in terms of the distances. SPARQL queries are generated with a graph pattern of a sequential path from given entity $r \in R$ to literal $\ell \in L$ which contains query term t, and predicates and resources between r and ℓ are fulfilled by free variables. Figure 3 illustrates a n-length graph pattern for entity r and query term t. Based on

Table 4: NDCG@k (k=10, 100). Model indicates task types of queries, and top-k indicates the selected k values (10 or 100). Each cell contains an NDCG@k value for corresponding condition. For each column, the best score is boldface and underlined. The most-left column lists the state-of-the-art and re-ranked versions of them by PPRSD (corresponding with *-ed names). Each group of rows corresponding with the state-of-the-art includes imp. row indicating the ratio of the improvement by PPRSD.

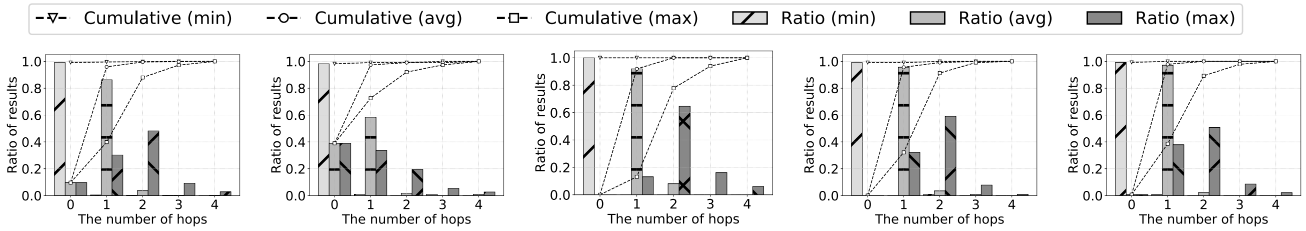
Model	SemSearch ES		INEX-LD		ListSearch		QALD-2		Total	
	@10	@100	@10	@100	@10	@100	@10	@100	@10	@100
BM25	.2497	.4110	.1828	.3612	.0627	.3302	.2751	.3366	.2558	.3582
BM25*	.2839	.4463	.2903	.3816	.2534	.3543	.2953	.3624	.2812	.3847
imp.	+13.70%	+8.59%	+58.81%	+5.65%	+304.15%	+7.30%	+7.34%	+7.66%	+9.93%	+7.40%
PRMS	.5340	.6108	.3590	.4295	.3684	.4436	.3151	.4026	.3905	.4688
PRMS*	.5388	.6162	.3590	.4295	.3684	.4436	.3151	.4026	.3913	.4698
imp.	+0.90%	+0.88%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	+0.20%	+0.21%
MLM-all	.5528	.6247	.3752	.4493	.3712	.4577	.3249	.4208	.4021	.4852
MLM-all*	.5578	.6303	.3752	.4493	.3712	.4577	.3249	.4208	.4030	.4863
imp.	+0.90%	+0.90%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	+0.22%	+0.23%
LM	.5555	.6475	.3999	.4745	.3925	.4723	.3412	.4338	.4182	.5036
LM*	.5606	.6529	.3999	.4745	.3925	.4723	.3413	.4338	.4191	.5046
imp.	+0.92%	+0.83%	0.00%	0.00%	0.00%	0.00%	+0.03%	0.00%	+0.22%	+0.20%
SDM	.5535	.6672	.4030	.4911	.3961	.4900	.3390	.4274	.4185	.5143
SDM*	.5564	.6718	.4030	.4912	.3961	.4902	.3394	.4274	.4191	.5152
imp.	+0.52%	+0.69%	0.00%	+0.02%	0.00%	+0.04%	+0.12%	0.00%	+0.14%	+0.17%
LM-ELR	.5554	.6469	.4040	.4816	.3992	.4845	.3491	.4383	.4230	.5093
LM-ELR*	.5608	.6518	.4040	.4816	.3992	.4847	.3491	.4383	.4240	.5103
imp.	+0.97%	+0.76%	0.00%	0.00%	0.00%	+0.04%	0.00%	0.00%	+0.24%	+0.20%
SDM-ELR	.5548	.6680	.4104	.4988	.4123	.4992	.3446	.4363	.4261	.5211
SDM-ELR*	.5577	.6716	.4105	.4988	.4129	.4999	.3449	.4364	.4271	.5218
imp.	+0.52%	+0.54%	+0.02%	0.00%	+0.15%	+0.14%	+0.09%	+0.02%	+0.23%	+0.13%
MLM-CA	.6247	.6854	.4029	.4796	.4021	.4786	.3365	.4301	.4365	.5143
MLM-CA*	.6249	.6895	.4029	.4798	.4020	.4786	.3365	.4301	.4361	.5150
imp.	+0.03%	+0.60%	0.00%	+0.04%	-0.02%	0.00%	0.00%	0.00%	-0.09%	+0.14%
BM25-CA	.5858	.6883	.4120	.5050	.4220	.5142	.3566	.4426	.4399	.5329
BM25-CA*	.6040	.7024	.4132	.5048	**.4302**	**.5181**	.3607	.4544	.4475	.5404
imp.	+3.11%	+2.05%	+0.29%	-0.04%	+1.94%	+0.76%	+1.15%	+2.67%	+1.73%	+1.41%
FSDM	.6521	.7220	.4214	.5043	.4196	.4952	.3401	.4358	.4524	.5342
FSDM*	.6549	.7269	.4214	.5044	.4196	.4951	.3401	.4359	.4527	.5350
imp.	+0.43%	+0.68%	0.00%	+0.02%	0.00%	-0.02%	0.00%	+0.02%	+0.07%	+0.15%
BM25F-CA	.6281	.7200	.4394	.5296	.4252	.5106	.3689	.4614	.4605	.5505
BM25F-CA*	.6444	**.7361**	**.4494**	**.5336**	.4288	.5166	**.3699**	**.4672**	**.4673**	**.5581**
imp.	+2.60%	+2.24%	+2.28%	+0.76%	+0.85%	+1.18%	+0.27%	+1.26%	+1.48%	+1.38%
FSDM-ELR	.6563	.7257	.4354	.5134	.4220	.4985	.3468	.4456	.4590	.5408
FSDM-ELR*	**.6572**	.7307	.4354	.5135	.4219	.4985	.3466	.4455	.4587	.5416
imp.	+0.14%	+0.69%	0.00%	+0.02%	-0.02%	0.00%	-0.06%	-0.02%	-0.07%	+0.15%

the pattern, *ASK* query (which is an indicator function query in SPARQL) is generated to examine such pattern exists. Following SPARQL query displays examples of generated ASK queries for distance 2.

```
ASK{ ⟨r⟩ ?p0 ?v0. ?v0 ?p1 ?v1.
     ?v1 ?p2 ?l. ?l bif:contains 't'.
     FILTER isLiteral(?l).}
```

This investigation measures the minimum distance which satisfies the ASK query corresponding with the distance. The procedure of this investigation is that: (1) given query q, relevant entity list A_q for q is obtained from the benchmark dataset; (2) parse q into set T_q of terms; (3) examine ASK queries from length 0 to maximum length (5 for this investigation) for each pair of relevant entity $r \in A_q$ and term $t \in T_q$; (4) as soon as the ASK query is satisfied, the distance is recorded; and (5) the obtained distances for each relevant entities are analyzed. Obtained distances for a relevant entity of a query may be different term by term. Therefore, this investigation analyses minimum distance, average distance and maximum distance for each relevant entity of a query. Consequently, these distances are individually gathered and calculate their averages to observe how long distances required to touch query terms from relevant entities.

Figure 2 showcases the analyzed distances with respect to tasks as well as with regardless of tasks (i.e., *Total*). In the figure, bars represent ratios of relevant entities having the number of hops (distances) to reach from query terms, and dashed lines express cumulative ratios of relevance entities. Three kinds of bars (light-gray and oblique stripe bars, gray and horizontal stripe bars, and black and crossing stripe bars) correspond



(a) Total (b) SemSearch ES (c) INEX-LD (d) ListSearch (e) QALD-2

Figure 2: The number of hops from relevant entities to query terms. Bars represent ratios of relevant entities having the number of hops (distances) to reach from query terms. Three kinds of bars (light-gray and oblique stripe bars, gray and horizontal stripe bars, and black and crossing stripe bars) correspond with minimum, average, and maximum distances, respectively. Dashed lines express cumulative ratios of relevance entities. Three kinds of lines (lines with triangles, those with circles, and those with squares) correspond with minimum, average, and maximum, respectively.

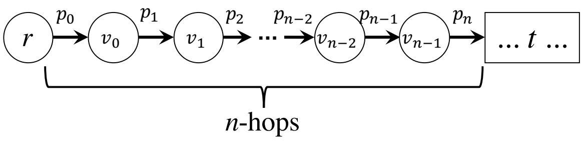


Figure 3: n-length path pattern generated for given entity r, query term t and distance n. Circular vertices are resources and a square is a literal containing t.

with minimum, average (rounded), and maximum distances. Similarly, three kinds of lines (lines with triangles, circles, and squares) correspond with minimum, average (rounded), and maximum.

Figure 2 indicates that at least one term is included in literals directly connected with relevant entities, and Figure 2(a) indicates that most of the relevant entities are reachable from query terms within two hops on average, however, in terms of maximum distances, still more than 10% relevant entities are not reachable within three hops. This fact answers the question *why recall@1000 is not perfect?* as some relevant entities are still not found by the query terms due to the smaller distances to construct entity documents. This phenomenon is also marked on individual tasks except SemSeach ES task, which is a simple tasks so that queries in the task are more directly explaining requiring entities than others.

5.2 Commonality of Tail Predicates of Paths

A simple solution for improving ranking qualities in terms of the previous investigation is top include literals within more hops (i.e., 3 or more), however, it is obvious that the solution incurs noisy entity documents by including unnecessary literals within larger hops. The number of reachable entities in G increases very quickly as the distance increases. Therefore, irrelevant entities contribute to entity documents.

An intuition to avoid this situation is to select "good" paths from an entity which include meaningful literals for the entity. A naïve extension is to find paths from an entity to "good" entities and to include their documents (suppose the same approach to the state-of-the-art) into the document of the entity. This paper wants to clarify there is any difference between *self-descriptive* literals and *supportive* literals for other entities. Self-descriptive literals explain well about target entities, while supportive literals explain supplemental facts about the target entities. Self-descriptive literals tend to be close to the targets, while supportive literals tend to relatively distant from the targets. Therefore, this investigation attempts to understand the differences of predicates with ending literals (called tail predicates) between shorter and longer paths. The investigation is done in the following procedure: gathers surveyed paths for each relevant entities using intermediate results of the previous investigation (Section 5.1), and analyzes the paths in terms of commonalities of the tail predicates. The commonalities are measured for different lengths (i.e., 1 to 4) of the tail predicate sequences by Jaccard index, $Jaccard(Y_i^r, Y_j^r) = \frac{|Y_i^r \cap Y_j^r|}{|Y_i^r \cup Y_j^r|}$, where Y_i^r is a set of i-length tail predicates of entity r and $|\cdot|$ is cardinality.

Table 5 display commonalities of tail predicates among different lengths of paths for different tail lengths. The results reveal that commonalities of tail predicates decrease as differences of path lengths increase. This fact indicates that literals reachable in different path lengths should select different tail predicates (e.g., `rdfs:label` is not always a good choice.). Due to the tremendous number of tail predicates, the detailed analysis on what kind tail predicates are preferable in particular path lengths is left for future work. Examples from rough analysis include `rdfs:label` and `rdfs:comment` for 1-length paths and `dbo:wikiPageWikiLinkText` for 5-length paths.

Table 5: Commonality (Jaccard index) of tail predicates of top-10 frequent paths from true results to query keywords. The numbers of top-most and left-most in the tables represent lengths of the paths. These tables show that only a part (less than 45%) of tail predicates which are related to basic documents of entities is shared with different lengths of paths.

(a) tail length = 1

	Path length				
	1	2	3	4	5
1	1.000	0.399	0.220	0.250	0.198
2	0.399	1.000	0.429	0.342	0.316
3	0.220	0.429	1.000	0.389	0.439
4	0.250	0.342	0.389	1.000	0.449
5	0.198	0.316	0.439	0.449	1.000

(b) tail length = 2

	Path length			
	2	3	4	5
2	1.000	0.205	0.250	0.190
3	0.205	1.000	0.325	0.316
4	0.250	0.325	1.000	0.449
5	0.190	0.316	0.449	1.000

(c) tail length = 3

	Path length		
	3	4	5
3	1.000	0.250	0.282
4	0.250	1.000	0.481
5	0.282	0.481	1.000

(d) tail length = 4

	Path length	
	4	5
4	1.000	0.449
5	0.449	1.000

5.3 Summary & Future Direction

Summary.

This section investigates relationship between paths and result entities in order to answer the question *Why recall@1000 is not 100% yet?* The answers of this investigation are 2-fold: (1) literals in distant paths are absent from documents; and (2) setting of tail predicates is universal for all lengths of paths. Although, the first problem is obvious, still the number of reachable entities in more than two hops is extraordinary large, therefore, constructing documents from longer paths is computationally expensive. Additionally, in the naïve approach, the generated documents may include large amount of not quite relevant facts to entities.

Future Directions.

To overcome the aforementioned problems, solutions lies on graph analytical approaches as Section 3 showcases their possibilities. A basic idea is to select appropriate reachable predicates and entities within two or more hops. To this end, graph analytical approaches (e.g., PageRank, Random Walks) can be good choices. As discussed in this paper, non-personalized PageRank and its families are not appropriate, meaning that global centralities do not help. Therefore, customizable graph analytical approaches such as ObjectRank [BHP04] and random walk with restart (RWR) [TFP08] are preferable. There are some preliminary works based on this idea, namely FORK [KOAK17], and RWRDoc [Kom18]. FORK has applies ObjectRank for entity search and it achieves better precision@k. While, RWRDoc has applies RWR for determining importances of reachable entities in terms of RWR scores and it slight improves in terms of NDCG@k. These indicates that graph analytical approaches still leave space for improvements.

6 Conclusion

This paper deals with entity search over Linked Data, analyzes the state-of-the-art in terms of recall@k, and reveals the possibilities of improvements of the state-of-the-art. Also, this paper indicates the feasibility of graph analytical approaches for improving the state-of-the-art by formulating as a re-ranking problem. For further improvements, this paper reports two investigations about relationship between paths to literals containing query terms and relevant entities to queries. Results of the investigations support the improvement possibilities of graph analytical approaches, and developing them is still left for future works. Consequently, this paper answers to the first question as *Yes, they are appropriate, but there is still an issue on matching entities in a graph.*

Acknowledgments.

This work was partly supported by JSPS KAKENHI Grant Number JP18K18056.

References

[BHB09] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.

[BHP04] Andrey Balmin, Vagelis Hristidis, and Yannis Papakonstantinou. ObjectRank: Authority-Based Keyword Search in Databases. In *VLDB 2004*, pages 564–575, 2004.

[Has18] Faegheh Hasibi. *Semantic Search with Knowledge Bases*. PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2018.

[Hav02] Taher H. Haveliwala. Topic-sensitive PageRank. In *WWW 2002*, pages 517–526, 2002.

[HBB16] Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. Exploiting Entity Linking in Queries for Entity Retrieval. In *ICTIR 2016*, pages 209–218, 2016.

[HNX+17] Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. DBpedia-Entity v2: A Test Collection for

Entity Search. In *SIGIR 2017*, pages 1265–1268, 2017.

[KOAK17] Takahiro Komamizu, Sayami Okumura, Toshiyuki Amagasa, and Hiroyuki Kitagawa. FORK: Feedback-Aware ObjectRank-Based Keyword Search over Linked Data. In *AIRS 2017*, pages 58–70, 2017.

[Kom18] Takahiro Komamizu. Learning Interpretable Entity Representation in Linked Data. In *DEXA 2018*, 2018. (to appear).

[KXC09] Jinyoung Kim, Xiaobing Xue, and W. Bruce Croft. A Probabilistic Retrieval Model for Semistructured Data. In *ECIR 2009*, pages 228–239, 2009.

[MC05] Donald Metzler and W. Bruce Croft. A Markov Random Field Model for Term Dependencies. In *SIGIR 2005*, pages 472–479, 2005.

[OC03] Paul Ogilvie and James P. Callan. Combining Document Representations for Known-item Search. In *SIGIR 2003*, pages 143–150, 2003.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, November 1999.

[PC98] Jay M. Ponte and W. Bruce Croft. A Language Modeling Approach to Information Retrieval. In *SIGIR 1998*, pages 275–281, 1998.

[PMZ10] Jeffrey Pound, Peter Mika, and Hugo Zaragoza. Ad-hoc Object Retrieval in the Web of Data. In *WWW 2010*, pages 771–780, 2010.

[RZ09] Stephen E. Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *FTIR*, 3(4):333–389, 2009.

[TFP08] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Random walk with restart: fast solutions and applications. *Knowl. Inf. Syst.*, 14(3):327–346, 2008.

[UNH$^+$17] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, Michael Röder, and Giulio Napolitano. 7th Open Challenge on Question Answering over Linked Data (QALD-7). In *ESWC 2017*, pages 59–69, 2017.

[WKC$^+$12] Qiuyue Wang, Jaap Kamps, Georgina Ramírez Camps, Maarten Marx, Anne Schuth, Martin Theobald, Sairam Gurajada, and Arunav Mishra. Overview of the INEX 2012 Linked Data Track. In *CLEF 2012 Evaluation Labs and Workshop*, 2012.

[ZKN15] Nikita Zhiltsov, Alexander Kotov, and Fedor Nikolaev. Fielded Sequential Dependence Model for Ad-Hoc Entity Retrieval in the Web of Data. In *SIGIR 2015*, pages 253–262, 2015.