J. Shanghai Jiao Tong Univ. (Sci.), 2018, 23(4): 568-576
https://doi.org/10.1007/s12204-018-1976-z

Querying Linked Data Based on Hierarchical Multi-Hop Ranking Model

LI Junxian [1,2]* (李俊娴), *WANG Wei*[2] (汪　卫), *WANG Jingjing*[3] (王晶晶)
(1. School of Computer, Jiangsu University of Science and Technology, Zhenjiang 212003,
Jiangsu, China; 2. School of Computer Science, Fudan University, Shanghai 200433, China;
3. Yangzhou Polytechnic College, Yangzhou 225009, Jiangsu, China)

Abstract: How to query Linked Data effectively is a challenge due to its heterogeneous datasets. There are three types of heterogeneities, i.e., different structures representing entities, different predicates with the same meaning and different literal formats used in objects. Approaches based on ontology mapping or Information Retrieval (IR) cannot deal with all types of heterogeneities. Facing these limitations, we propose a hierarchical multi-hop language model (HMPM). It discriminates among three types of predicates, descriptive predicates, out-associated predicates and in-associated predicates, and generates multi-hop models for them respectively. All predicates' similarities between the query and entity are organized into a hierarchy, with predicate types on the first level and predicates of this type on the second level. All candidates are ranked in ascending order. We evaluated HMPM in three datasets, DBpedia, LinkedMDB and Yago. The results of experiments show that the effectiveness and generality of HMPM outperform the existing approaches.

Key words: hierarchical multi-hop ranking model (HMPM), Linked Data, language model
CLC number: TP 391　　**Document code:** A

0 Introduction

With the development of Linked Data, querying becomes challenging because of the large number and heterogeneity of datasets. 1024 (http://datahub.io/stats) Linked Data sets are published and the number is still increasing. And three types of heterogeneity exist. For example, Fig. 1 contains three graph models of "Braveheart" (red nodes) in DBpedia (DBpedia. http://wiki.dbpedia.org/About), LinkedMDB (Linked Movie DataBase, http://www. linkedmdb.org) (L: file/38074) and Yago (Yago.http://www.mpi-inf.mpg.de/yago-naga/yago/) respectively. Entities are shown as circles, predicates as directed edges, literal objects as rectangles. Source and destination of an edge represent subject and object respectively.

(1) Different predicates have the same meaning. "starring"(DBpedia), "actor"(LinkedMDB) and "actedIn"(Yago) show the same meaning. "title", "name" and "HasPreferred-Name" are virtually interchangeable.

(2) Entities' structures are various. In DBpedia and Yago, actor of "Braveheart" is "D: Sophie Marceau" by obtaining the object of corresponding predicate. However, in LinkedMDB, the further information of "L: actor/29743" should be retrieved to identify the actor. Besides, Yago stores actor within person (actedIn), but not in movie. Predicates with reversed direction (colored green) should be retrieved. To deal with above differences, two hops information in DBpedia, three hops in LinkedMDB and four hops information should be retrieved when querying "Braveheart"(L: file/38074).

(3) Literal formats of an entity are different. "Mel Gibson", "Gibson, Mel" and "mel gibson" denote the same person in Fig. 1.

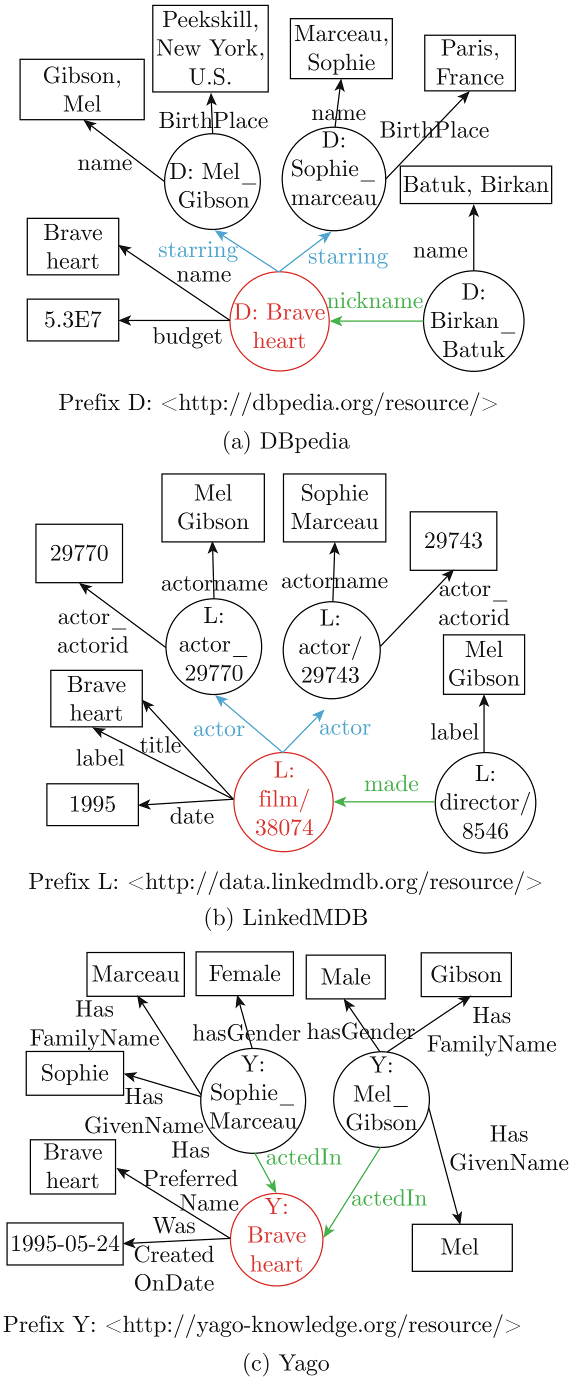
A SPARQL (SPARQL Protocol and RDF Query Language) query: select ?s where {?s actor "Sophie Marceau". ?s actor "Mel Gibson".}, which aims to find the movie that was acted by "Mel Gibson" and "Sophie Marceau". It cannot be executed directly in Linked Data because of those heterogeneity discussed previously.

Weaknesses of previous approaches. There are two main directions of querying Linked Data, but both have limitations:

(1) Database approaches. On-line link traversal without mapping[1-5] can only handle exact query. Off-line ontology or schema mapping[6-9] can only match different predicates in different datasets.

(2) Information Retrieval (IR) approaches. Keyword augmentation model[11] is not suitable for Linked Data

Received date: 2018-04-16
***E-mail:** lijunxian_just@163.com



(a) DBpedia

(b) LinkedMDB

(c) Yago

Fig. 1 Data heterogeneity in Linked Data

consisting of many large datasets. Unstructure Entity Model (USEM), Structured Entity Model (SEM), Hierarchical Entity Model (HEM)[12] and Entity Relevance Model (ERM)[13] cannot handle structural heterogeneity.

Facing these limitations, we suggest that ideal Linked Data query should be executed smoothly based on the model with generality. The challenge is how to generate a model that can handle all heterogeneity.

We propose a hierarchical multi-hop ranking model (HMPM). It discriminates among three types of predicates, descriptive, out-associated, and in-associated predicate, then generates models by computing probability distributions of terms contained in the direct and indirect objects of each predicates (section 1). Query is modeled in the same way. The relevant entities are ranked in ascending order according to similarities, which organizes all similarities of predicates into hierarchical model (section 2). Our approach can tackle the three heterogeneous problems at the same time (section 3).

1 Multi-Hop Language Model of Predicates

Our target is to model a query and entities based on language model to support querying heterogenous Linked Data. We define three types of predicates (descriptive, out-associated and in-associated predicate), and model them respectively. Table 1 gives the notations used in this paper.

Descriptive Predicate Predicate uses literal as its object. We use the same model in Ref. [13] for this type of predicate. Let $P(t|a, \mathrm{de})$ presents the probability of t occurring in objects of predicate a belonging to the type of de, and it can be defined as follows:

$$P(t|a, \mathrm{de}) = \frac{\displaystyle\sum_{o \in \mathrm{obj}(a)} n(\mathrm{label}(o), t)}{\displaystyle\sum_{o \in \mathrm{obj}(a)} |\mathrm{label}(o)|}, \tag{1}$$

where $n(\mathrm{label}(o), t)$ denotes the count of term t contained in the object o, $|\mathrm{label}(o)|$ is the total number of terms contained in o and the sum goes over all objects of predicate a.

In Fig. 1, black edges are descriptive predicates when modeling “Braveheart” (L: film/38074).

Out-Associated Predicate Predicate uses the targeted entity as its subject. In Fig. 1, blue edges are out-associated predicates when modeling “Braveheart” (L: film/38074). Let $P(t|a, \mathrm{os})$ presents the probability of t occurring in objects of predicate a belonging to the type of os, it can be defined as follows:

$$P(t|a, \mathrm{os}) = P(t|\mathrm{obj}(a), \mathrm{obj}(\mathrm{Dnb}(\mathrm{obj}(a)))) = \frac{\displaystyle\sum_{o \in \mathrm{obj}(a)} n(\mathrm{label}(o), t) + \sum_{o' \in \mathrm{obj}(a)} \sum_{a' \in \mathrm{Dnb}(o')} \sum_{o \in \mathrm{obj}(a')} n(\mathrm{label}(o), t)}{\displaystyle\sum_{o \in \mathrm{obj}(a)} |\mathrm{lable}(o)| + \sum_{o' \in \mathrm{obj}(a)} \sum_{a' \in \mathrm{Dnb}(o')} \sum_{o \in \mathrm{obj}(a')} |\mathrm{label}(o)|}, \tag{2}$$

Table 1 Notations

Notation	Meaning	Example (Fig. 1(b))
R, r	Entity	L: film/38074, L: director/8546, L: actor/29770, L: actor/29743
a	Predicate	title, actor, date, actor_actorid, actorname, made
a_t={de, os, is}	Set of type of predicate: descriptive(de), out-associated(os), in-associate (is)	
type (a)	Type of a	type(label)=de, type(actor)=os, type(made)=is
$A_{\text{de}}, A_{\text{os}}, A_{\text{is}}$	Set of predicates with one type	A_{de}={date, title, label, actor_actorid, actor_actorname}, A_{os}={actor}, A_{is}={made}
sbj(a)	Set of subjects of a	sbj(made)=L: director/8546
obj(a)	Set of objects of a	obj(actor)={L: actor/29970, L: actor/29743}
label(x)	Character string of x	label(L: actor/29970)=actor/29970 label(Braveheart)=Braveheart
t	Term	1995, Mel, film

tm(a)	Set of all term with a	tm(made)={director, 8546, Mel, Gibison}
		tm(title)={Braveheart}

Dnb(r)	Set of descriptive predicates of r	Dnb(L: film/38074)={title, label, date}

where it is estimated by using the labels of direct objects (obj(a)) and indirect literal ones (ojb(Dnb(obj(a)))). $n(\text{label}(o), t)$ are the count of term t contained in the label of object o. $|\text{label}(o)|$ is the total number of terms contained in the label of object o. The sum in the first component goes over all direct objects of predicate a. The sum in the second component goes over all indirect literal objects of predicate a.

In Fig. 1(b), "actor" is an out-associated predicate, and its direct objects are:

obj(actor)={L:actor/29770, L: actor/29743}.

The descriptive predicates of "L: actor/29770" and "L: actor/29743" are:

$$\text{Dnb(L : actor/29770)} = \text{Dnb(L : actor/29743)} = \{\text{actor\_actorid}, \text{actorname}\}.$$

The indirect literal objects of "actor" are the objects of "actor acotid" and "actname":

$$(\text{Dnb}(\text{obj}(\text{actor})) = \{38074, \text{MelGibson}, 29743, \text{SophieMarceau}\}.$$

In-Associated Predicate Predicate uses the targeted entity as its object. In Fig. 1, green edges are in-associated predicates when modeling "Braveheart" (L: film/38074). Let $P(t|a, \text{is})$ presents the probability of t occurring in objects of predicate a belonging to the type of is, it can be defined as follows:

$$P(t|a, \text{is}) = P(t|\text{sbj}(a), \text{obj}(\text{Dnb}(\text{obj}(a)))) = \frac{\sum\limits_{s \in \text{sbj}(a)} n(\text{label}(s), t) + \sum\limits_{s' \in \text{sbj}(a)} \sum\limits_{a' \in \text{Dnb}(s')} \sum\limits_{o \in \text{obj}(a')} n(\text{label}(o), t)}{\sum\limits_{s \in \text{sbj}(a)} |\text{lable}(s)| + \sum\limits_{s' \in \text{sbj}(a)} \sum\limits_{a' \in \text{Dnb}(s')} \sum\limits_{o \in \text{obj}(a')} |\text{label}(o)|}, \quad (3)$$

where the language model of a is estimated using the labels of subjects (sbj(a)) and indirect literal objects (obj(Dnb(sbj(a)))). $n(\text{label}(s), t)$ and $n(\text{label}(o), t)$ are the count of term t contained in the label of subject s and object o respectively, label(s) and label(o) are the total numbers of terms contained in the label of subjects s and object o respectively. The sum in the first component goes over all subjects of predicate a. The sum in the second component goes over all indirect literal objects of predicate a.

In Fig. 1(c), "actedIn" is an in-associated predicate, and its subjects are:

$$\text{sbj(actedIn)} = \{\text{Y : Mel\_Gision}, \text{Y : Sophie\_Marceau}\}.$$

The descriptive predicates of "Y: Mel Gibson" and "Y: Sophie Marceau" are:

$$\text{Dnb(Y : Mel\_Gibsion)} = \text{Dnb(Y : Sophie\_Marceau)} = \{\text{HasGivernName}, \text{HasFamilyName}, \text{HasGender}\}.$$

The indirect literal objects of “actedIn” are the objects of “HasGivernName”, “HasFamilyName” and “HasGender”:

$$\text{Dnb}(\text{sbj}(\text{actedIn})) = \{\text{Sophie}, \text{Marceau}, \text{female}, \text{Mel}, \text{Gebson}, \text{made}\}.$$

Multi-Hop Predicate Model (MPM) Based on those three models, MPM of the given predicate is:

$$\text{MPM}(a) = \prod_{t \in \text{tm}(a)} P(t|a, \text{type}(a)). \quad (4)$$

We calculated the MPM of “L: film/38074” in Fig. 1(b), and the results are displayed in Table 2.

Table 2 MPM of “Braveheart” in LinkedMDB in Fig. 1(b)

Type	Predicates	$t: P(t\|a, \text{type}(a))$
Descriptive	Title	Braveheart: 1
	Label	Braveheart: 1
	Date	1995: 1
Out-associated	Actor	film: 0.2, 29770: 0.2, 29743: 0.2, Mel: 0.1, Gibson: 0.1, Sophie: 0.1, Marceau: 0.1
In-associated	Made	director: 0.25, 8546: 0.25, Mel: 0.25, Gibsion: 0.25

2 Hierarchical Multi-Hop Models for Ranking

We propose a HMPM to rank candidate entities. It calculates the similarity between two predicates from the query and entity by Kullback-Leibler divergence (KL-divergence), and organizes all similarities within the query into a hierarchy of two level and then ranks candidate entities in ascending order.

Similarity Between Two Predicates We use KL-divergence, which is used to measure the “distance” between two probability distributions, to calculate the similarity. Let a_q and a_r be two predicates from the query q and the entity r in dataset. The similarity between them is:

$$\text{Sim}(a_q, a_r) = \text{KL}(\text{MPM}(a_q) \| \text{MPM}(a_r) = \sum_{t \in \text{tm}(a_q)} P(t|a_q, \text{type}(a_q)) \times \log \frac{P(t|a_q, \text{type}(a_q))}{\lambda P(t|a_r, \text{type}(a_q)) + (1-\lambda) P(t|D)}, \quad (5)$$

where $P(t|a_q, \text{type}(a_q))$ is computed by Eq. (1), (2) or (3) depending on the predicate type $\text{type}(a_q)$; the denominator of the fraction is the probability of term t in predicate a_r which is smoothed by its probability in the entire dataset D; the parameter λ controls the influence of smoothing. The sum goes over all terms in predicate a_q. It is crucial to smooth the probabilities, since $P(t|a_r, \text{type}(a_q)) = 0$ while t does not exist in the predicate a_r. λ, whose effect on performance has been studied extensively for IR tasks, was set to 0.9.

HMPM To discriminate among three predicate types, all predicate similarities should be organized into a hierarchy of two levels, with predicate types on the first level and predicates of that type on the second level when calculating the similarity between the query and entity. Inspired by the work in Ref. [12], we propose our hierarchical ranking model, and the graphical representation is shown in Fig. 2.

The HMPM is generated as following:

$$\text{HMPM}(q, r) = \text{Sim}(q, r) = \frac{1}{\min\{|a_q|, |a_r|\}} \left[\sum_{\text{tp} \in a_t} \left(\sum_{a_q \in A^q_{\text{tp}}, a_r \in A^r_{\text{tp}}} \text{Sim}(a_q, a_r) \times P(a_q|\text{tp}, q) \right) P(\text{tp}|q) \right], \quad (6)$$

where the square bracket component is hierarchical model based on $\text{Sim}(a_q, a_r)$. The denominator is the minimum number of predicates contained in query and entity. Similarity based on KL-divergence measures “distance”. So the more number of predicates is calculated, the larger “distance” is. If the number of predicates in the entity is smaller than that in query, the similarity will be bias. We use the denominator to average the similarity.

We now describe how to estimate $P(a|\text{tp}, q)$ and $P(\text{tp}|q)$. In our model, $\text{tp} \in a_t, a_t = \{\text{de}, \text{os}, \text{is}\}$. $P(a|\text{tp}, q)$ captures the importance of the predicate a conditioned on its type and q, and it relies on popularity and type. We claim that in a query, the more popular the predicate is, the more important it is. It is estimated as follows:

$$P(a|\text{tp}, q) = \frac{n(a|\text{tp}, q)}{|A^q_{\text{tp}}|}, \quad (7)$$

where $n(a|\text{tp}, q)$ is the number of a belonging to type in q and $|A^q_{\text{tp}}|$ is the total number of predicates with type tp in the query q.

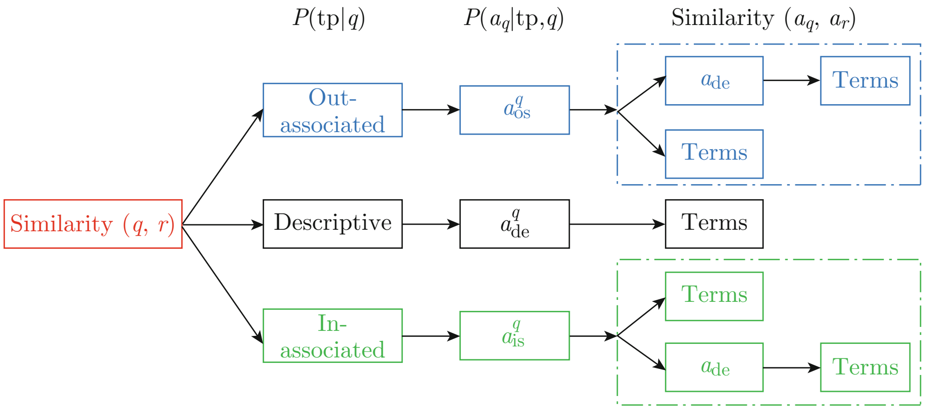


Fig. 2 HMPM

$P(\mathrm{tp}|q)$ allows us to set importance of predicate type for each query. For simplicity, we use these weights: $P(\mathrm{de}|q) = 1$, $P(\mathrm{os}|q) = 0.9$, $P(\mathrm{is}|q) = 0.9$.

The right two columns in Table 3 are the weights of predicates and types. The results of the given query are returned to the user in ascending order.

Table 3 MPM of query 10 in Table 5

Type	Predicates	$t : P(t\|a, \mathrm{type}(a))$	$P(a\|\mathrm{type}(a), q)$	$P(\mathrm{type}(a)\|q)$
Descriptive	Star	Mel: 0.5, Gibsion: 0.25 Sophie: 0.25, Marceau: 0.25	0.67	1
	Director	Mel: 0.5, Gibsion: 0.5	0.33	1

Searching Strategies For a given query, the number of comparisons is nm_rm_q. n is the number of entities, m_r is the number of predicates within an entity, and m_q is the number of predicates in query. Although the m_r and m_q are relatively small, query process still iterates over many times because n is very large. Thus, it is time-consuming. Strategies reducing the number of iterations are required. We use LDA (Latent Dirichlet Allocation) algorithm[16] to cluster entities to reduce the search space. LDA is also based on probability distribution of terms, so the clusters by it are beneficial for our models. We choose the cluster which has the largest overlapping predicates between the query and entity to compute similarity. The effects of the number of clusters (C) will be discussed in section 3.

3 Evaluation

This section presents our experimental results, which demonstrate that our model effectively tackles heterogeneity when querying Linked Data. Results reported about $\mathrm{HMPM_L}$ in the following are obtained by setting $C = 10$.

3.1 Datasets

Our experiments were conducted on 3 Linked Data sets: DBpedia core, LinkedMDB, and Yago. DBpedia is a structured representation of Wikipedia. We extracted triples containing predicates with "http://dbpedia.org/property/" to constitute dataset named "DBpedia core". Link-edMDB is a source about movies. Yago is derived from Wikipedia, WordNet and GeoNames. We use the Core datasets, containing core facts of Yago. Table 4 gives the statistics about each dataset. As there is a big overlap about movies and other entities related with movies among these three datasets, they offer sufficient duplicates for evaluating our model.

Table 4 Dataset statistics

Dataset	Entities	Distinct Predicates
DBpedia core	9.1×10^6	3.86×10^4
LinkedMDB	5.0×10^5	222
Yago core	1.0×10^7	125

3.2 Query Sets and Ground Truth

To find query-relevant entities in three heterogeneous datasets, we designed 10 queries by using arbitrary vocabulary named as $\mathrm{QS_a}$. Then rewrote them by using our 3 datasets vocabularies and adhering to corresponding structure representation, named as $\mathrm{QS_d}$, $\mathrm{QS_l}$, $\mathrm{QS_y}$. We executed $\mathrm{QS_d}$ in DBpedia core, $\mathrm{QS_l}$ in LinkedMDB and $\mathrm{QS_y}$ in Yago. Then the results of them capture the ground truth. Table 5 lists the 10 queries in $\mathrm{QS_a}$.

Table 5 QS$_{\mathrm{a}}$

No.	Query
1	Select ?s where {?s star “Mel Gibsion”.}
2	Select ?s where {?s actor “Sophie Marceau”. ?s music “James Horner”. }
3	Select ?s where {?s star “Mel Gibsion”. ?s direct “Mel Gibsion”.}
4	Select ?s, ?o1, ?o2 where {?s star “Sophie Marceau”. ?o1 music ?o2. ?o2 won “Grammy”. }
5	Select ?s where {?s star “Mel Gibsion”. ?s date “1995”. }
6	Select ?s where {?s star “Sophie Marceau”. ?s star “Mel Gibsion”. ?s direct “Mel Gibsion”. ?s music “James Horner”. }
7	Select ?s where {?s director ?o. ?o name “Rainer Werner Fassbinder”. }
8	Select ?s where {?s publishdata “1995”.}
9	Select ?s where {?s production company “Pixar”.}
10	Select ?s where (?s star “Sophie Marceau”. ?s star “Mel Gibsion”.?s director “Mel Gibsion”.}

3.3 Systems

In experiments, we submitted each query to three datasets. Similarity is calculated by hierarchical ranking model in section 2. We compared our two models(HMPMR, HMPML) against other works(USEM[12], ERM[13]) as discussed previously in introduction. HMPMR computes similarity without search strategy. HMPML combines LDA with HMPM. This version applies LDA algorithm to clustering entities based on predicates and objects with different “topics”, then executes query under a smaller search space.

All experiments were carried on a server with Intel Xeon 2.13 GHz CPU, 64 GB RAM, 1 TB hard drive and Windows Server 2008 and all approaches were implemented by Python 2.7.

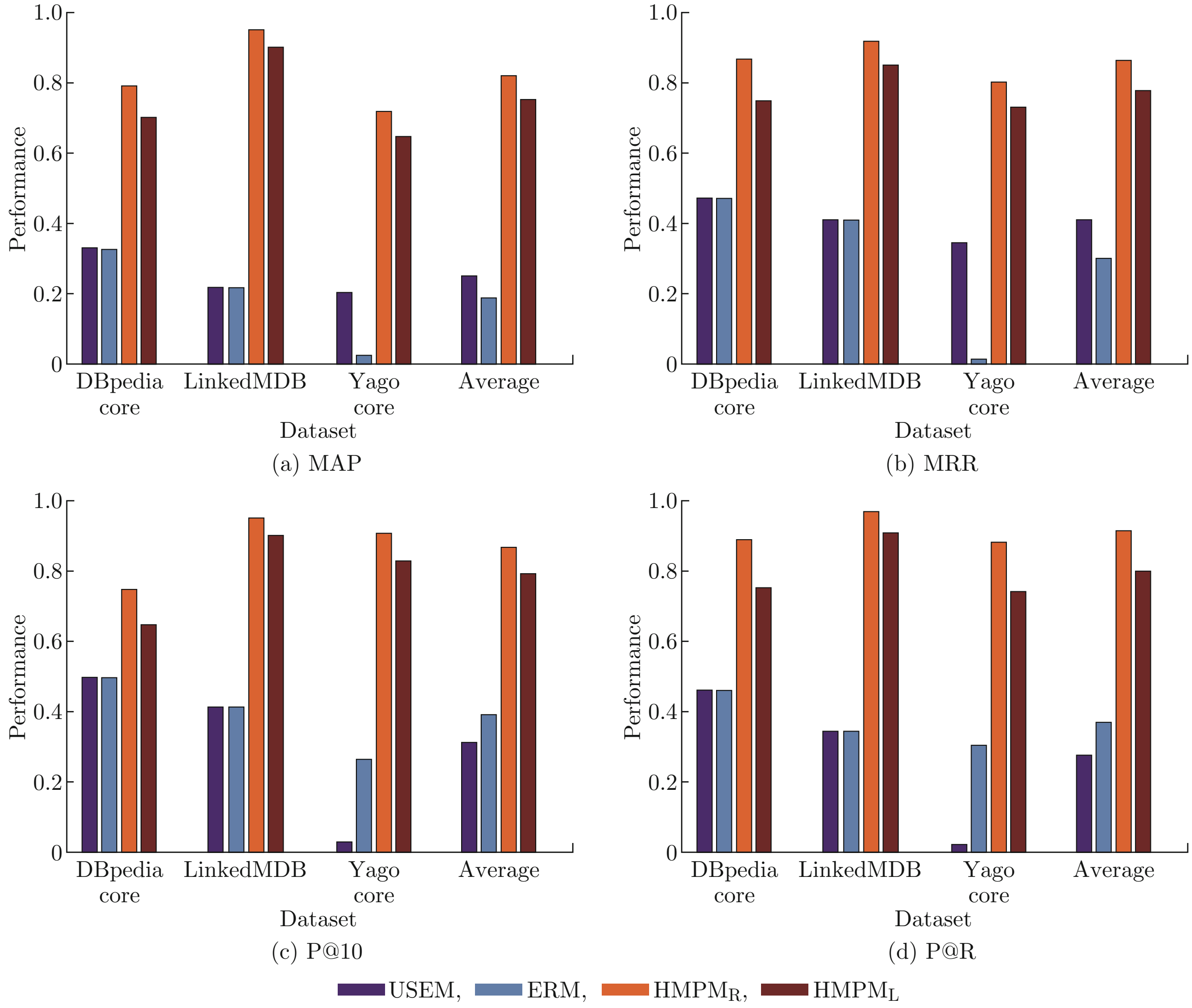
3.4 Query Effectiveness

Firstly, Table 6 gives the top 3 movies returned by each model when executing query 2 from QS$_a$ (Table 5) and the bold ones are the right results. The answer is “Braveheart” in all datasets. The ranks of “Braveheart” using USEM are 7, 92, 76 in three datasets respectively and ERM ranked it in 9, 92, 76. They are wrong results. We can verify our analysis that USEM and ERM cannot deal with all heterogeneity, in contrast HMPMR and HMPML work well on all three datasets.

Table 6 Results of query 2 using USEM, ERM, HMPMR and HMPML

Dataset	Rank	USEM	ERM	HMPMR	HMPML
DBpedia	1	Martin Riggs	Martin Riggs	**Braveheart**	**Braveheart**
	2	Female Agents	Female Agents	Firelight	Fanfan
	3	Saw III	Saw III	Fanfan	The man without a face
	*	**Braveheart**(7)	**Braveheart**(7)		
LinkedMDB	1	45 227	45 227	**38 074**	**38 074**
	2	39 937	39 937	83 123	98 194
	3	11 471	11 471	98 194	8 621
	*	**38 074**(92)	**38 074**(92)		
Yago	1	Waking ned	Waking ned	**Braveheart**	**Braveheart**
	2	Gangs of new York	Gangs of new York	Pour sacha	The man without a face
	3	Uncommon valor	Uncommon valor	La boum	pacalypto
	*	**Braveheart**(76)	**Braveheart**(76)		

Secondly, we use the standard IR evaluation metrics: mean average precision (MAP), mean reciprocal rank (MRR), precision at rank 10 (P@10) and R-precision (P@R). We retrieve the top one hundred entities, rank them, and compute the metrics based on the top thirty entities returned by each model. We designed nine different retrieval settings, QS$_{\mathrm{a}}$ was submitted to 3 datasets, QS$_{\mathrm{d}}$ was submitted to LinkedMDB and Yago, QS$_{\mathrm{l}}$ was submitted to DBpedia and Yago, and QS$_{\mathrm{y}}$ was submitted to DBpedia and LinkedMDB. The results of those metrics grouped on datasets are shown in Fig. 3. Both HMPMR and HMPML outperform USEM and ERM across all metrics and datasets. Observing the different datasets, both HMPMR and HMPML perform the best on LinkedMDB, because LinkedMDB has entities from simple domain. Both DBpedia and Yago have entities from multiple domains. Query performance are improved remarkably. As the search space is reduced in HMPML, the query performance of HMPML is not good as HMPMR, but is still better than ERM and



(a) MAP

(b) MRR

(c) P@10

(d) P@R

Fig. 3 Query performance in DBpedia, LinkedMDB and Yago

USEM.

Figure 4 is the interpolated precision across the recall level in three datasets, which shows the robustness of query performance. It can be observed that precisions of HMLML are decreased notably at recall levels above 70%, 80% and 60% in DBpeda, LinkedMDB and Yago respectively. HMLMR is fairly stable over different recall levels in LinkedMDB. In DBpedia and Yago, the precisions of HMLMR are decreased at recall levels above 70% and 60% respectively.

3.5 Runtime Performance

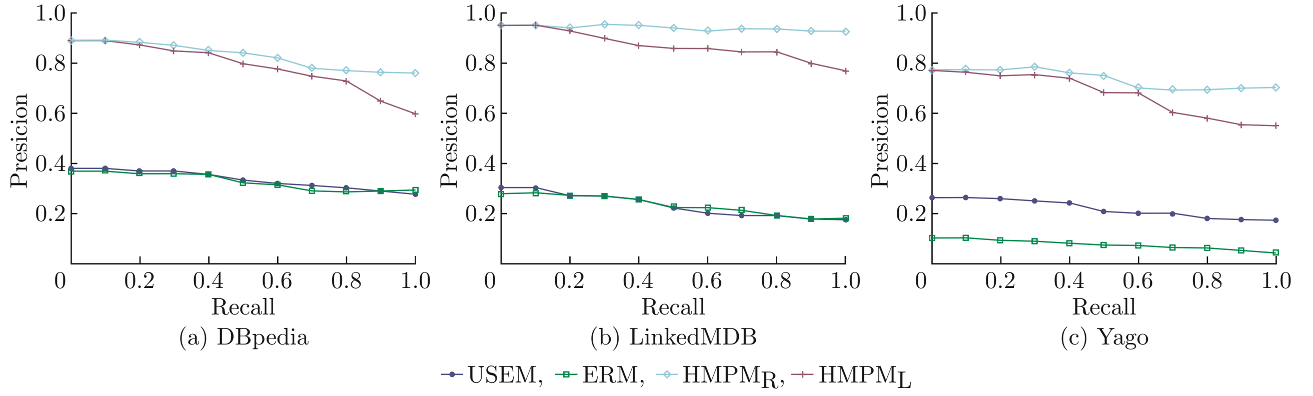
To analyze the effectiveness of search strategy, we measured query execution time for ERM, HMPMR and HMPML on all datasets. The time of HMPML covers the LDA step. We use the ERM on each dataset as the baseline, the ratios of the time cost to it are listed in Table 7. Terms contained in indirect objects are required in our model, thus computing similarity between two predicates by our models needs more time than ERM. In HMPML, we clustered the entities by using LDA algorithm to reduce the search space. The execution time of HMPML listed in table by setting $C = 1$, $C = 2$, $C = 10$. In fact, HMPMR is HMPML when $C = 1$. The performance can be improved by increasing the cluster number C, but the precision is decreasing.

Table 7 Comparison of time cost

	DBpedia	LinkedMDB	Yago
ERM	1	1	1
HMPM_R $(C = 1)$	7.71	1.48	2.40
HMPM_L $(C = 2)$	3.77	0.67	1.12
HMPM_L $(C = 10)$	0.58	0.11	0.21

3.6 Parameter Analysis

HMPML relies on the parameter C for clustering. We analyze the effect of C in terms of MAP for the nine



(a) DBpedia

(b) LinkedMDB

(c) Yago

Fig. 4 Precision-recall curves

retrieval settings by setting C as 1, 2, 10. We observed that the precision is decreasing when C is increasing (Fig. 5).

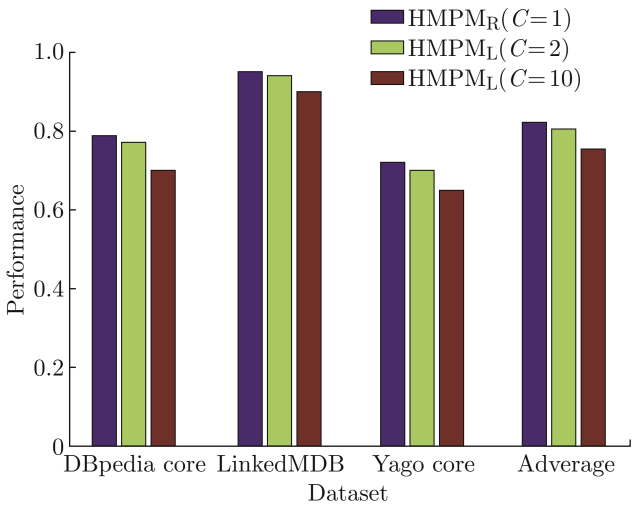


Fig. 5 Clusters C

4 Related Work

We have discussed related work in introduction. References [1-5] retrieved answers by exact matching, so they do not consider the heterogeneity. Schema mapping approaches[6,7] were studied well in heterogeneous database field and ontology mapping approaches[8-10] were studied in semantic field. Both are not suitable for Linked Data because of the large number of heterogenous datasets. IR approaches[11-13] based on language model are proposed recently. The main intuition in those approaches is generating probability models for a query and entities. They can deal with different predicates and different literal formats, but not the structure representation heterogeneity.

5 Conclusion

We have proposed a new ranking model for querying heterogeneous Linked Data. We have introduced three types of heterogeneities among Linked Data and discussed the existing approaches based on database or IR techniques. But those approaches cannot handle all types of heterogeneities. Our hierarchical ranking model is used to find and rank entities that are relevant to the given query. Our approach can handle all heterogeneities. Combining the LDA algorithm, our model can improve the query performance. Extensive experiments conducted with three real-world datasets show the effectiveness of our approach and verify the generality and scalability of our ranking model. In this paper, we only focus on star query, so we will extend this approach to permit more general query, like path query and hybrid query.

References

[1] HARTIG O, BIZER C, FREYTAG J C. Executing SPARQL queries over the web of Linked Data [C]//*Proceedings of the 8th International Semantic Web Conference*. Chantilly, VA, USA: Springer, 2009: 293-309.

[2] LADWIG G, TRAN T. Linked Data query processing strategies [C]//*Proceedings of the 9th International Semantic Web Conference*. Shanghai, China: Springer, 2010: 453-469.

[3] HARTIG O. Zero-knowledge query planning for an iterator implementation of link traversal based query execution [C]//*Proceedings of the 8th Extended Semantic Web Conference*. Heraklion, Crete, Greece: Springer, 2011: 154-169.

[4] HARTH A, HOSE K, KARNSTEDT M, et al. Data summaries for on-demand queries over Linked Data [C]//*Proceedings of the 19th International Conference on World Wide Web*. Raleigh, NC, USA: DBLP, 2010: 411-420.

[5] PHAM M D, BONCZ P. Exploiting emergent schemas to make RDF systems more efficient [C]//*Proceedings*

of the 15th International Sematic Web Conference. Kobe, Japan: Springer, 2016: 463-479.

[6] RAHM E, BERNSTEIN P A. A survey of approaches to automatic schema matching [J]. *The International Journal on Very Large Data Bases*, 2001, **10**(4): 334-350.

[7] DOAN A H, HALEVY A Y. Semantic-integration research in the database community: A brief survey [J]. *American Association for Artificial Intelligence*, 2005, **26**(1): 83-94.

[8] DUAN S, FOKOUE A, SRINIVAS K. One size does not fit all: Customizing ontology alignment using user feedback [C]// *Proceedings of the 10th International Semantic Web Conference*. Shanghai, China: Springer, 2010: 177-192.

[9] HU W, QU Y Z. Falcon-AO: A practical ontology matching system [J]. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2008, **6**(3): 237-239.

[10] ZHOU X, GAUGAZ J, BALKE W T, et al. Query relaxation using malleable schemas [C]//*Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. Beijing, China: ACM, 2007: 545-556.

[11] ELBASSUONI S, RAMANATH M, SCHENKEL R, et al. Language-model-based ranking for queries on RDF-graphs [C]//*Proceedings of the 18th ACM Conference on International and Knowledge Management*. Hong Kong, China: ACM, 2009: 977-986.

[12] NEUMAYER R, BALOG K, NØRVÅG K. On the modeling of entities for ad-hoc entity search in the web of data [C]//*Proceedings of the 34th European Conference on IR Research*. Barcelona, Spain: Springer, 2012: 133-145.

[13] HERZIG D M, TRAN T. Heterogeneous web data search using relevance-based on the fly data integration [C]//*Proceedings of the 21st International Conference on World Wide Web*. Lyon, France: ACM, 2012: 141-150.

[14] PONTE J M, CROFT W B. A language modeling approach to information retrieval [C]//*Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Melbourne, Australia: ACM, 1998: 275-281.

[15] OGILVIE P, CALLAN J. Hierarchical language models for XML component retrieval [C]//*Proceedings of the 3rd International Conference on Initiative for the Evaluation of XML Retrieval*. Dagstuhl Castle, Germany: Springer, 2004: 224-237.

[16] BLEI D M, NG A Y, JORAN M I. Latent dirichlet allocation [J]. *Journal of Machine Learning Research*, 2003, **3**: 993-1022.