Entity Set Expansion via Knowledge Graphs

Xiangling Zhang
Renmin University of China
Beijing, China
zhangxiangling@ruc.edu.cn

Yueguo Chen*
Renmin University of China
Beijing, China
chenyueguo@ruc.edu.cn

Jun Chen
Renmin University of China
Beijing, China
chenjun2013@ruc.edu.cn

Xiaoyong Du
Renmin University of China
Beijing, China
duyong@ruc.edu.cn

Ke Wang
Simon Fraser University
Burnaby, Canada
wangk@cs.sfu.ca

Ji-Rong Wen
Renmin University of China
Beijing, China
jrwen@ruc.edu.cn

ABSTRACT

The entity set expansion problem is to expand a small set of seed entities to a more complete set of similar entities. It can be applied in applications such as web search, item recommendation and query expansion. Traditionally, people solve this problem by exploiting the co-occurrence of entities within web pages, where latent semantic correlation among seed entities cannot be revealed. We propose a novel approach to solve the problem using knowledge graphs, by considering the deficiency (e.g., incompleteness) of knowledge graphs. We design an effective ranking model based on the semantic features of seeds to retrieve the candidate entities. Extensive experiments on public datasets show that the proposed solution significantly outperforms the state-of-the-art techniques.

KEYWORDS

Knowledge Graph; Entity Set Expansion; Entity Search

1 INTRODUCTION

The entity set expansion (ESE) problem is to find similar entities to a given small set of seed entities. For example, given the seed entities *Barack_Obama*, *John_Kennedy* and *Franklin_Roosevelt*, we may expect to find entities such as *Bill_Clinton* and *Jimmy_Carter* because they are all US presidents from the Democratic Party. It can be widely used in many applications such as web search (search by examples), item recommendation and query expansion [10].

Traditionally, people solve this problem using a web corpus (e.g., SEAL [9] and BBR [2]), by evaluating the similarities between candidate entities and the seeds based on their surrounding contexts within the corpus. Entities that co-occur more frequently with the seeds are likely to have higher similarities. Unfortunately, these methods are time-consuming since both web crawling and entity extraction are costly. Moreover, common features shared by the seeds cannot be revealed by these methods. There have been a number of path-based similarity measures [7, 8] to evaluate the similarity between a pair of entities in knowledge graphs (KGs) which can be adopted to solve the ESE problem. Metzger et al. [6] propose a solution to ESE called QBEES based on the common features shared by the seeds. It however ignores the deficiency (incompleteness) of the KGs which affects the precision. The association rule mining (ARM) algorithm [1] can also be adapted to solve the ESE problem. However, it lacks of an effective ranking model, which cannot distinguish the importance of the common features shared among seeds.

Knowledge graphs such as DBpedia and Freebase are widely used in the fields of web search and question answering. The facts in KGs are typically represented by triples ($< s, p, o >$) describing the properties of the subjects as well as the relations among entities. We utilize p^- to represent the inverse relation of the predicate p. The whole KGs can be represented as directed and labeled graphs. Figure 1 shows an example of a KG. Although huge, exsting KGs are still incomplete. For example, 71% of people in Freebase lack place of birth information [4].

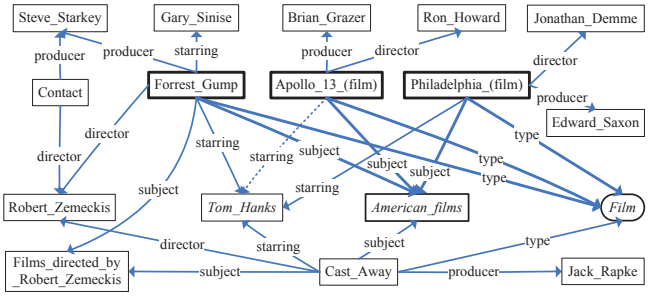

Figure 1: A running example, where a dashed triple $<$ *Apollo_13_(film)*, *starring*, *Tom_Hanks* $>$ is missed.

In this paper, We propose a ranking model to effectively evaluate the similarity of entities to a small number of given seeds, according to the semantic correlations (called *k-relaxed* common semantic features) among entities in knowledge graphs. The model is designed to handle the incompleteness of knowledge graphs. We use an example of Figure 1 to give a quick view of the idea. Suppose the user's intention is *Tom Hanks movies where he plays a leading role* and he issues a query of three seeds, *Forrest_Gump*, *Apollo_13_(film)*, and *Philadelphia_(film)*. We may find that two seeds have the same predicate *starring* to the same entity *Tom_Hanks*, which implies that *Tom_Hanks* played a role in them. We therefore can apply the label *starring* and the entity *Tom_Hanks* as a 1-relaxed common

*Yueguo Chen is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR '17, August 07–11, 2017, Shinjuku, Tokyo, Japan.

© 2017 ACM. 978-1-4503-5022-8/17/08…$15.00

DOI: http://dx.doi.org/10.1145/3077136.3080732

feature among the given seeds. Based on this common feature, we can find some other entities (e.g. *Cast_Away*) sharing the same common feature as the majority of the seeds, and it therefore can be applied for ranking candidate entities.

2 COMMON SEMMANTIC FEATURE

Definition 2.1 (Semantic Feature). A semantic feature $\pi = e_a : P$ in a KG $\mathcal{K}$ is composed of an anchor entity e_a, and a sequence of labels $P = l_1/l_2/\ldots/l_n$.

A semantic feature (SF) is used to represent a common feature shared by a set of target entities. For example, to describe the movies where *Tom_Hanks* played a role, we can apply the SF π_1=*Tom_Hanks*:$starring^-$. For another example, if we want to define people who directed movies where *Tom_Hanks* played a role, the SF can then be written as: π_2=*Tom_Hanks*:$starring^-/director$, which has two predicates to define the relation. The length of a SF is the number of labels (predicates) in P. It can be larger than one when P is a tandem of several predicates (e.g., π_2), although the cases of length one (a direct relation) are used more often.

If an entity e has a relation P with the anchor entity e_a, we say that e is a target entity of π=e_a:P, which is denoted as $e \models \pi$. The set of target entities of a SF π=e_a:P is denoted as $E(\pi) = \{e|e \models \pi\}$. For a given set of seed entities, we may find some SFs whose target entities contain all those seed entities. They are defined as the common semantic features (shorted as CSFs) of the seeds. We use $\Phi(Q)$ to denote the set of CSFs for the seeds in a query Q. For example, for the seed entity set $\{Forrest\_Gump, Apollo\_13\_(film), Philadelphia\_(film)\}$, SFs π_3=*Film*:$type^-$ and π_4= $American\_films$:$subject^-$ are their CSFs.

To overcome the deficiency of KGs, we relax the definition of CSFs as follows.

Definition 2.2 (k-relaxed CSF). A semantic feature π is a k-relaxed CSF to a set of m entities in Q, if $|E(\pi) \cap Q| \geq m - k$.

A k-relaxed CSF π requires that at least $m-k$ entities of the seeds are target entities of π, i.e., $|E(\pi) \cap Q| \geq m - k$. We use k-CSF to denote a k-relaxed CSF, and $\Phi^k(Q)$ to represent the set of k-relaxed CSFs of the seed set Q. To solve the ESE problem using KGs, we need to follow two steps to rank entities based on the proposed CSF: 1) compute the set of CSFs ($\tilde{\Phi}(Q)$) according to the given query Q; 2) retrieve and rank the candidate entities (target entities excluding the seeds) satisfying the detected CSFs in $\tilde{\Phi}(Q)$.

As discussed above, due to the limits of the coverage of KGs, there may be a very small number (or even not any) of CSFs (of length one) shared by all the seeds in Q, we therefore apply the relaxation of CSFs by allowing some seeds not satisfying the CSFs. Moreover, the length of CSFs can be extended to be larger than one, so as to include more CSFs indicating indirect common features. However, the relaxation and extension of CSFs will of course generate more false positives of common features that may not be desired by the user. In addition, more CSFs will reduce the search performance as well. The selection of CSFs for ranking entities, therefore has to be carefully designed.

Let $\Phi^k_h(Q)$ be the set of k-CSFs of Q whose length is no more than h, where $k \geq 0$ and $h \geq 1$. In our solution, we apply the union of two sets for ranking entities, i.e., $\tilde{\Phi}(Q) = \Phi^k_1(Q) \bigcup \Phi^0_h(Q)$, where $\Phi^k_1(Q)$ includes k-CSFs of length one, and $\Phi^0_h(Q)$ includes CSFs whose length is extended up to h. However, those CSFs longer than one will not be relaxed to avoid generating too many false positive CSFs.

3 RANKING MODEL

The ranking model of entities is designed as follows:

$$r(e) = \sum_{\pi \in \tilde{\Phi}(Q) \wedge e \models \pi} d(\pi) * r(\pi, Q) \tag{1}$$

It is basically an aggregation of the score of each CSF $\pi \in \tilde{\Phi}(Q)$ that e satisfies, which is further evaluated as the product of two components $d(\pi)$, and $r(\pi, Q)$, where $d(\pi)$ is the discriminability of π, and $r(\pi, Q)$ is the relevance of π to Q.

3.1 Discriminability of CSFs

It is likely that many CSFs can be found from KGs, although only some of them are useful for finding similar entities of seeds. For example, to characterize the seeds, π_1 is more specific than π_3 because $|E(\pi_1)| \ll |E(\pi_3)|$. We therefore need a measure to evaluate the discriminability of CSFs on finding similar entities. Intuitively, the discriminability of π is then defined as:

$$d(\pi) = \frac{1}{|E(\pi)|} \tag{2}$$

Larger $|E(\pi)|$ means that entities in $E(\pi)$ are more loosely correlated in terms of the constraint of π. It therefore has a smaller discriminability of the relevant entities.

3.2 Relevance of CSFs

The relevance of a CSF π to the query Q, is evaluated as:

$$r(\pi, Q) = \prod_{e \in Q} p(e, \pi) \tag{3}$$

where $p(e, \pi)$ is the probability of e satisfying π. For $e \models \pi$, $p(e, \pi)$ is naturally evaluated as 1. However, for a relaxed k-CSF, there can be at most k seeds that do not satisfy π, which may be caused by the deficiency of the KGs. We therefore need to evaluate $p(e, \pi)$ for those seeds that do not satisfy π. Borrowing the idea of collaborative filtering in recommendation systems, we evaluate $p(e, \pi)$ by considering the likelihood of e satisfying similar CSFs of π.

$$p(e, \pi) = \begin{cases} 1 & \text{if } e \models \pi \\ \frac{\sum_{\pi' \in \Psi(\pi)} I(e, \pi') w(\pi', \pi)}{\sum_{\pi' \in \Psi(\pi)} w(\pi', \pi)} & \text{otherwise} \end{cases}$$

where $I(e, \pi') = 1$ if $e \models \pi'$, otherwise $I(e, \pi') = 0$; $w(\pi', \pi) = \frac{|E(\pi') \cap E(\pi)|}{|E(\pi)|}$, which determines the weight of π'; $\Psi(\pi) = \{\pi' | \pi' = e_a : P_x\} \bigcup \{\pi' | \pi' = e_x : P\}$ with $\pi = e_a : P$ and the length of P_x is one. Obviously, the set of similar CSFs of π, $\Psi(\pi)$, is derived by substituting the anchor entity (from e_a to any other e_x) or the path (from P to any other P_x of length one) of $\pi = e_a : P$ respectively.

4 EXPERIMENTS

The DBpedia v3.9 is applied as the KGs of our experiments. Two test datasets are used in our study. QALD [5], the Question Answering over Linked Data campaign, aims to answer natural language questions using linked data sources. After removing the redundant

Table 1: Comparison on the QALD dataset

solution	seeds	p@5	p@10	p@20	MRR	R-pre
SEAL	2	.377	.290	.208	.550	.269
BBR	2	.340	.305	.245	.446	.263
LDSD	2	.147	.122	.100	.264	.113
QBEES	2	.507	.400	.320	.654	.369
ARM	2	.503	.422	.322	.662	.377
ESER	2	**.547**$^{\bullet}_{*}$	**.460**$^{\bullet}_{*}$	**.372**$^{\bullet}_{*}$	**.699**$^{\bullet}_{*}$	**.457**$^{\bullet}_{*}$
SEAL	3	.453	.363	.267	.591	.340
BBR	3	.393	.335	.276	.505	.298
LDSD	3	.170	.143	.127	.270	.131
QBEES	3	.557	.440	.362	.688	.423
ARM	3	.550	.468	.372	.665	.446
ESER	3	**.613**$^{\bullet}_{*}$	**.498**$^{\bullet}_{*}$	**.387**$^{\bullet}_{*}$	**.773**$^{\bullet}_{*}$	**.501**$^{\bullet}_{*}$
SEAL	4	.420	.350	.270	.539	.354
BBR	4	.363	.312	.270	.526	.302
LDSD	4	.197	.163	.138	.308	.153
QBEES	4	.557	.453	.362	.668	.452
ARM	4	.527	.430	.348	.716	.420
ESER	4	**.613**$^{\bullet}_{*}$	**.502**$^{\bullet}_{*}$	**.392**$^{\bullet}_{*}$	**.801**$^{\bullet}_{*}$	**.525**$^{\bullet}_{*}$
SEAL	5	.410	.317	.247	.535	.352
BBR	5	.350	.323	.273	.515	.304
LDSD	5	.173	.145	.127	.282	.153
QBEES	5	.520	.428	.348	.638	.449
ARM	5	.503	.418	.342	.665	.426
ESER	5	**.563**$^{\bullet}_{*}$	**.465**$^{\bullet}_{*}$	**.381**$^{\bullet}_{*}$	**.726**$^{\bullet}_{*}$	**.515**$^{\bullet}_{*}$
SEAL	mix	.447	.347	.249	.592	.335
BBR	mix	.373	.328	.262	.477	.298
LDSD	mix	.183	.155	.137	.292	.141
QBEES	mix	.517	.408	.328	.626	.412
ARM	mix	.537	.443	.337	.646	.433
ESER	mix	**.633**$^{\bullet}_{*}$	**.510**$^{\bullet}_{*}$	**.403**$^{\bullet}_{*}$	**.799**$^{\bullet}_{*}$	**.559**$^{\bullet}_{*}$

Table 2: Comparison on the INEX dataset

solution	seeds	p@5	p@10	p@20	MRR	R-pre
SEAL	2	**.412**$_{*}$	**.388**$_{*}$	**.331**$_{*}$	.542$_{*}$	**.327**$_{*}$
BBR	2	.304	.248	.213	.418	.209
LDSD	2	.219	.200	.153	.461	.166
QBEES	2	.392	.338	.288	.556	.282
ARM	2	.319	.287	.231	.496	.244
ESER	2	.400$_{*}$	.383$_{*}$	.287$_{*}$	**.551**$_{*}$	.304$_{*}$
SEAL	3	.462$_{*}$	**.433**$_{*}$	**.354**$_{*}$	.547$_{*}$	**.377**$_{*}$
BBR	3	.292	.246	.211	.470	.208
LDSD	3	.227	.210	.172	.401	.184
QBEES	3	.362	.317	.255	.532	.256
ARM	3	.281	.260	.216	.435	.224
ESER	3	**.500**$_{*}$	.415$_{*}$	.311$_{*}$	**.684**$^{\star}_{*}$	.340$_{*}$
SEAL	4	.423$_{*}$	.383$_{*}$	.319$_{*}$	.530$_{*}$	.339$_{*}$
BBR	4	.277	.235	.210	.447	.213
LDSD	4	.246	.235	.176	.408	.179
QBEES	4	.362	.312	.234	.451	.229
ARM	4	.292	.256	.216	.493	.222
ESER	4	**.504**$^{\star}_{*}$	**.446**$^{\star}_{*}$	**.341**$^{\star}_{*}$	**.633**$^{\star}_{*}$	**.376**$^{\star}_{*}$
SEAL	5	.377	.340	.284	.418	.311
BBR	5	.300	.250	.208	.530	.219
LDSD	5	.300	.292	.203	.535	.208
QBEES	5	.246	.215	.169	.342	.169
ARM	5	.315	.267	.211	.484	.239
ESER	5	**.492**$^{\star}_{*}$	**.433**$^{\star}_{*}$	**.336**$^{\star}_{*}$	**.629**$^{\star}_{*}$	**.381**$^{\star}_{*}$
SEAL	mix	.473$_{*}$	.398$_{*}$	.305$_{*}$	.644$_{*}$	.330$_{*}$
BBR	mix	.323	.277	.221	.504	.251
LDSD	mix	.262	.227	.164	.496	.204
QBEES	mix	.423	.371	.276	.591	.299
ARM	mix	.350	.304	.239	.535	.273
ESER	mix	**.515**$^{\star}_{*}$	**.440**$^{\star}_{*}$	**.350**$^{\star}_{*}$	**.701**$^{\star}_{*}$	**.409**$^{\star}_{*}$

topics, we get a dataset of 60 topics from QALD-2, QALD-3 and QALD-4. In INEX-XER 2009 (shorted as INEX with 55 topics) [3], a topic contains a natural language question asking for a list of entities. In addition, it also provides several seed entities as the examples of the desired entities. We use the label ESER (for testID) to indicate that the test is under the default setting. All significant tests are conducted using a one-tailed t-test at a significance level of $p = 0.05$.

4.1 An Overall Comparison

We first test the performance of the compared solutions using 5 groups of different numbers of seeds. Note that the mix group contains topics whose numbers of seeds are between 2 to 5. In general, LDSD [7] performs worse than the others on both datasets, which shows that a simple semantic distance approach on entities of KGs is far from judging effective semantic correlations among entities. Generally, ESER performs the best on almost all the test cases, with two exceptions beaten by SEAL [9] on the INEX dataset when the query contains only 2 or 3 seeds. SEAL benefits a lot from the usage of Google search engine. The way of using frequent item sets on predicate-object pairs serves the purpose of finding common features of seeds. However, the lack of an effective ranking model causes that ARM [1] performs worse than ESER. The notation $_{*}$ denotes significant difference over ARM, the notation $^{\bullet}$ denotes significant difference over QBEES [6] in Table 1, and the notation * denotes significant difference over SEAL in Table 2.

When looking into the impact of the number of seeds m on the performance of the different solutions, we find that most solutions perform worst when $m = 2$, which means that there are not enough seeds to distinguish the common features shared among the seeds. When m is enlarged from 2 to 5, the performance of SEAL and ESER basically increases. However, the growth rate is not significant when $m > 3$. For ESER, it benefits more from the enlargement of m on the INEX dataset than on the QALD dataset. This is reasonable because more seeds help ESER to discover more CSFs in the INEX dataset which are more implicit than those in QALD.

4.2 Effectiveness of The Ranking Model

Two components of ESER affect the performance of its ranking model: $d(\pi)$ and $r(\pi, Q)$. This experiment is designed to look into the performance of individual components by varying the overall ranking model. When a component is not applied in the ranking model, we simply set it 1. The results of the tests on two datasets are shown in Table 3 and Table 4 respectively. We apply 4 variations of the 2 components, with none of them applied as the baseline (in this case, candidate entities are ranked simply based on the number of k-CSFs they satisfy). Note that this experiment is conducted over the mix of QALD and the mix of INEX datasets individually. The results show that the 2 individual components can improve the search performance over the baseline approach (the first row of the two tables). The best performance is achieved when both components are applied (ESER), which is exactly the proposed ranking model.

Table 3: Alternative ranking models on the QALD

testID	$d(\pi)$	$r(\pi, Q)$	p@5	p@10	p@20	MRR	R-pre
q1	1	1	.493	.403	.326	.695	.399
q2	Eqn. 2	1	.560∗	.452	.359	.739	.453
q3	1	Eqn. 3	.550∗	.442∗	.350∗	.771∗	.462∗
ESER	Eqn. 2	Eqn. 3	**.633**∗	**.510**∗	**.403**∗	**.799**∗	**.559**∗

Table 4: Alternative ranking models on the INEX

testID	$d(\pi)$	$r(\pi, Q)$	p@5	p@10	p@20	MRR	R-pre
i1	1	1	.435	.350	.271	.661	.309
i2	Eqn. 2	1	.454	.410∗	.313∗	.689	.362
i3	1	Eqn. 3	.496∗	.427∗	.319∗	**.725**∗	.365∗
ESER	Eqn. 2	Eqn. 3	**.515**∗	**.440**∗	**.350**∗	.701	**.409**∗

Table 5: Strategies of picking CSFs on QALD

testID	$\tilde{\Phi}(Q)$	p@5	p@10	p@20	MRR	R-pre
q5	$\Phi_1^0(Q)$	.560	.462	.371	.663	.498
q6	$\Phi_1^1(Q)$	.593	.478	.383	.719∗	.525
q7	$\Phi_1^2(Q)$	.613∗	.490	.384	.769∗	.534
q8	$\Phi_1^3(Q)$	.623∗	.493∗	.386	.783∗	.537
q9	$\Phi_1^0(Q) \bigcup \Phi_2^0(Q)$	.583	.482∗	.394∗	.714∗	.527∗
q10	$\Phi_1^1(Q) \bigcup \Phi_2^0(Q)$	.607∗	.498∗	.401∗	.739∗	.549∗
q11	$\Phi_1^1(Q) \bigcup \Phi_2^1(Q)$	.640∗	.520∗	**.418**∗	.789∗	.548∗
q12	$\Phi_1^2(Q) \bigcup \Phi_2^0(Q)$	.623∗	.507∗	.402∗	.786∗	.556∗
q13	$\Phi_1^2(Q) \bigcup \Phi_2^1(Q)$	.640∗	.515∗	.408∗	.801∗	.553∗
q14	$\Phi_1^2(Q) \bigcup \Phi_2^2(Q)$	.643∗	.518∗	.408∗	.793∗	.547∗
ESER	$\Phi_1^3(Q) \bigcup \Phi_2^0(Q)$	.633∗	.510∗	.403∗	.799∗	**.559**∗
q16	$\Phi_1^3(Q) \bigcup \Phi_2^1(Q)$	.647∗	.517∗	.410∗	**.812**∗	.556∗
q17	$\Phi_1^3(Q) \bigcup \Phi_2^2(Q)$	.650∗	.520∗	.410∗	.804∗	.551∗
q18	$\Phi_1^3(Q) \bigcup \Phi_2^3(Q)$	**.657**∗	**.523**∗	.412∗	.812∗	.552∗

Table 6: Strategies of picking CSFs on INEX

testID	$\tilde{\Phi}(Q)$	p@5	p@10	p@20	MRR	R-pre
i5	$\Phi_1^0(Q)$	.458	.392	.299	.578	.349
i6	$\Phi_1^1(Q)$	.500	.433∗	.328∗	.672∗	.387∗
i7	$\Phi_1^2(Q)$	.492	.429∗	.332∗	.697∗	.390∗
i8	$\Phi_1^3(Q)$	.512	.448∗	.344∗	.717∗	.396∗
i9	$\Phi_1^0(Q) \bigcup \Phi_2^0(Q)$	.485∗	.412	.324	.611	.381∗
i10	$\Phi_1^1(Q) \bigcup \Phi_2^0(Q)$	.519∗	.442∗	.347∗	.675∗	.408∗
i11	$\Phi_1^1(Q) \bigcup \Phi_2^1(Q)$	**.538**∗	**.473**∗	.345∗	.696∗	.418∗
i12	$\Phi_1^2(Q) \bigcup \Phi_2^0(Q)$	.508	.437∗	**.350**∗	.701∗	.409∗
i13	$\Phi_1^2(Q) \bigcup \Phi_2^1(Q)$	.527∗	.469∗	.349∗	.721∗	.421∗
i14	$\Phi_1^2(Q) \bigcup \Phi_2^2(Q)$	.519∗	.469∗	.347∗	.720∗	.418∗
ESER	$\Phi_1^3(Q) \bigcup \Phi_2^0(Q)$	.515∗	.440∗	.350∗	.701∗	.409∗
i16	$\Phi_1^3(Q) \bigcup \Phi_2^1(Q)$	.531∗	.471∗	.349∗	**.721**∗	**.421**∗
i17	$\Phi_1^3(Q) \bigcup \Phi_2^2(Q)$	.519∗	.469∗	.347∗	.720∗	.418∗
i18	$\Phi_1^3(Q) \bigcup \Phi_2^3(Q)$	.527∗	.471∗	.346∗	.720∗	.416∗

4.3 Impacts of Selecting CSFs

ESER has two parameters that determine the set $\tilde{\Phi}(Q)$ of CSFs used for retrieving and ranking entities, and therefore affect the search performance. One is the parameter k used in discovering k-CSFs. The other is the parameter h used for constraining the length of CSFs. In this experiment, we test the impacts of these two parameters using the mix of QALD and the mix of INEX. The results are shown in Table 5 and Table 6. The significant tests are compared with the baselines (q5 and i5) where $k = 0$ and $h = 1$. For q5∼q8 and i5∼i8, we set $h = 1$, and enlarge k from 0 to 3. According to the results, the performance basically increases when k is enlarged, showing the effectiveness of k-relax CSFs for picking CSFs when the length of CSFs is limited to one.

When studying the impacts of relaxing 2-hop CSFs, we find that it may slightly reduce the search performance (e.g., q13 and q14 v.s. q12) on the QALD dataset. However, for INEX dataset, a small relaxation ($k = 1$) of 2-hop CSFs is helpful. This is also because INEX has a lower mapping quality and the relaxation does help to retrieve more CSFs. Considering that the relaxation of 2-hop CSFs often incurs more false positive CSFs, and therefore drops the search performance, we apply $\Phi_1^3(Q) \bigcup \Phi_2^0(Q)$ as the default setting of $\tilde{\Phi}(Q)$ for picking CSFs.

5 CONCLUSIONS

In this paper, we address the problem of entity set expansion by using KGs. We propose a concept called common semantic feature, to describe the common features shared by the seed entities, as the basis of discovering and ranking candidate entities. Through extensive experimental studies, we find that the proposed solution is very suitable for ESE topics which have good coverage of entities and predicates (relations) in the KGs. Even for those topics that do not have good information coverage in KGs, our noise-resistant solution may also work by discovering some common semantic features shared by the seeds.

6 ACKNOWLEDGMENTS

This work is supported by the National Science Foundation of China under grant (No. 61472426 and 61432006), 863 key project under grant No. 2015AA015307, the open research program of State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Science (No. CARCH201510), the ECNU-RUC-InfoSys Joint Data Science Lab, and a gift from Tencent.

REFERENCES

[1] Ziawasch Abedjan and Felix Naumann. 2013. Improving RDF Data Through Association Rule Mining. *Datenbank-Spektrum* 13, 2 (2013), 111–120.
[2] Krisztian Balog, Marc Bron, and Maarten de Rijke. 2011. Query modeling for entity search based on terms, categories, and examples. *ACM Trans. Inf. Syst.* 29, 4 (2011), 22.
[3] Gianluca Demartini, Tereza Iofciu, and Arjen P. de Vries. 2009. Overview of the INEX 2009 Entity Ranking Track. In *INEX*. 254–264.
[4] Xin Dong and Evgeniy Gabrilovich et al. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *KDD '14, New York, USA, August 24-27, 2014*. 601–610.
[5] Vanessa Lopez, Christina Unger, Philipp Cimiano, and Enrico Motta. 2013. Evaluating question answering over linked data. *J. Web Sem.* 21 (2013), 3–13.
[6] Steffen Metzger, Ralf Schenkel, and Marcin Sydow. 2014. Aspect-Based Similar Entity Search in Semantic Knowledge Graphs with Diversity-Awareness and Relaxation. In *WI and IAT*. 60–69.
[7] Alexandre Passant. 2010. dbrec - Music Recommendations Using DBpedia. In *ISWC*. 209–224.
[8] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. *PVLDB* 4, 11 (2011), 992–1003.
[9] Richard C. Wang and William W. Cohen. 2009. Automatic Set Instance Extraction using the Web. In *ACL*. 441–449.
[10] Jinxi Xu and W. Bruce Croft. 1996. Query Expansion Using Local and Global Document Analysis. In *SIGIR 1996*. 4–11.