计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(11):46−53 [doi: 10.15888/j.cnki.csa.008170]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

知识图谱的候选实体搜索与排序①

沈航可, 祁志卫, 张子辰, 岳　昆

(云南大学 信息学院, 昆明 650500)
通讯作者: 祁志卫, E-mail: maryqizhiwei@ynu.edu.cn

摘　要: 根据给定查询实体与知识图谱 (Knowledge Graph, KG) 中其他实体的相关程度对实体进行排序, 是相关实体搜索的重要支撑技术. 实体间的相关性不仅体现在 KG 中, 还体现在快速产生的 Web 文档中. 现有的方法主要根据 KG 来计算实体间的相关度, 但 KG 无法及时地反映真实世界中快速演化的知识, 导致计算结果不够客观. 因此, 本文首先基于 TransH 模型提出一种候选实体搜索算法, 通过分析实体在不同关系超平面中的语义表示来针对不同关系选择候选实体. 为了提高候选实体排序的准确性, 提出实体无向带权图模型 (Entity Undirected Weighted Graph, EUWG), 通过量化查询实体与候选实体在 Web 文档和 KG 中反映出的相关性, 从而准确地对候选实体进行排序. 实验结果表明, 本文的方法能够在大规模 KG 中准确地搜索候选实体并对其正确排序.

关键词: 知识图谱; 相关实体搜索; 表示学习; 无向带权图; 相关度

引用格式: 沈航可,祁志卫,张子辰,岳昆.知识图谱的候选实体搜索与排序.计算机系统应用,2021,30(11):46–53. http://www.c-s-a.org.cn/1003-3254/8170.html

Candidate Entity Search and Ranking of Knowledge Graph

SHEN Hang-Ke, QI Zhi-Wei, ZHANG Zi-Chen, YUE Kun

(School of Information Science & Engineering, Yunnan University, Kunming 650500, China)

Abstract: Ranking entities according to the relevance degree between the given entity and other entities in a Knowledge Graph (KG) is critical for related entity search. The relevance between entities is not only reflected in the KG but also the rapidly generated Web documents. In existing methods, the relevance degree is mainly calculated from the KG, which cannot reflect the knowledge rapidly evolving in the real world, and thus effective results cannot be obtained. Therefore, in this study, we first propose an algorithm for searching candidate entities on the basis of the TransH model by analyzing the semantic representation of entities in hyperplanes of different relations. To improve the precision of ranking candidate entities, we propose an Entity Undirected Weighted Graph (EUWG) model by quantifying the relevance between searched and candidate entities reflected in Web documents and KG. Experimental results show that the proposed method can precisely search and rank the candidate entities in the large-scale KG.

Key words: Knowledge Graph (KG); related entity search; representation learning; undirected weighted graph; relevance degree

1　引言

知识图谱 (Knowledge Graph, KG)[1] 作为实体关系的语义网络, 在相关实体搜索的应用中至关重要, 是搜索引擎的重要支撑技术[2]. 基于 KG 的相关实体搜索旨在根据给定的实体, 在 KG 中搜索与此实体相关的候选实体集合, 并按照候选实体与查询实体间的相关度

① 基金项目: 云南省万人计划“青年拔尖人才”计划 (C6193032); 云南大学“东陆学者”计划
Foundation item: Fund for Distinguished Young Scholars of Yunnan Province (C6193032); Cultivation Project of Donglu Scholar of Yunnan University
收稿时间: 2021-01-28; 修改时间: 2021-02-26; 采用时间: 2021-03-11; csa 在线出版时间: 2021-10-22

对候选实体进行排序并返回结果, 以提高用户的搜索体验. 事实上, 随着互联网的高速发展, Web 文档快速产生, 反映了现实世界不断演化的知识, 与 KG 中的知识共同描述了实体间的相关关系. 因此, 如何有效地表示实体在 KG 和 Web 文档中的关系信息, 进而准确地搜索与给定实体相关的候选实体, 并对候选实体进行排序, 对提升相关实体搜索的准确性具有重要意义. 虽然现有方法能够有效地获取相关实体, 减少用户搜索时需要过滤的无用信息, 但仍存在如下挑战:

(1) 与实体相连的不同关系能够表示实体不同的语义[3,4], 因此, 需要一种能够有效表示不同关系中实体的语义并准确搜索候选实体的方法.

(2) 由于 Web 文档与 KG 共同描述了实体间的相关关系, 为了准确地对候选实体进行打分排序, 需要一种能够根据实体在 Web 文档与 KG 中的关系信息来计算候选实体与查询实体间相关度的方法.

针对挑战 (1), 现有方法主要根据查询实体的邻居节点来搜索候选实体, 如 Huang 等[5] 使用与查询实体直接相连的实体作为候选实体集, Reinanda 等[4] 获取以查询实体为中心的 k 阶子图, 并基于子图的路径信息搜索候选实体. 上述方法在小规模 KG 中表现尚可, 而当 KG 规模较大时, 需要搜索的候选实体会出现在查询实体的邻居实体集外, 导致无法正确搜索到候选实体. 对此, 现有的表示学习方法[5–7] 将高维、复杂的 KG 映射到低维的向量空间中, 进而降低在大规模 KG 上的计算开销. 为了更加有效地搜索候选实体, 本文基于 TransH 模型[7] 提出候选实体搜索算法, 首先去除对查询实体不重要的关系, 降低搜索的时间代价. 然后通过 KG 的嵌入向量计算出实体在各关系对应超平面上的投影, 作为不同关系下实体的语义表示. 由于候选实体与查询实体有共同的语义特征[2], 因此, 为了有效地搜索候选实体, 我们根据实体的语义相似度对各超平面中的投影进行聚类, 进而得到与查询实体有共同语义特征的候选实体.

针对挑战 (2), 现有方法大多基于 KG 来计算实体相关度, 例如, Milne 等[8] 提出了 WLM 方法, 基于 KG 中实体所对应 Wikipedia 页面的超链接完成实体间的相关度计算. Ponza 等[9] 提出了 TSF (Two-Stage Framework) 方法, 利用 KG 实体间的连接关系构建带权有向图, 并基于 CoSimRank 算法[10] 来计算实体间的相关度. 这些算法能反映 KG 中实体间的相关性, 但由于现有 KG 的知识仍不完整[11], 导致计算结果不够准确. 对此, Yamada 等[12] 通过将描述实体的词汇和 KG 中的实体共同映射到向量空间, 以计算实体间的相关性. 该方法虽能将词汇与 KG 相结合来发现实体间的相关性, 但在映射过程中会损失 KG 实体间的关系信息, 导致计算结果不够准确. 因此, 为了更准确地计算查询实体与候选实体间的相关度, 我们提出实体无向带权图模型 (Entity Undirected Weighted Graph, EUWG). 首先, 以查询实体与候选实体作为图中节点, 基于查询实体与候选实体间的相关关系来构造无向边. 然后, 通过量化实体在 KG 向量空间和 Web 文档中体现出的相关性, 计算 EUWG 边上的权重, 得到查询实体与候选实体相互间的相关度, 并基于该模型提出一个候选实体打分函数, 通过遍历 EUWG 中实体间的路径计算候选实体的分数, 完成候选实体的排序.

最后, 使用 Wikidata 数据集, 对所提出的方法进行了实验测试和性能分析, 与现有的候选实体搜索算法和实体相关度计算模型进行比较, 验证了本文提出方法的有效性.

2 候选实体搜索

2.1 查询实体关系选择

定义 1. KG 是由实体和关系组成的有向图, 表示为 $G_{kg}=(E, R)$, 其中, $E=\{e_1, e_2, \cdots, e_n\}$ 为实体集合, $R=\{r_1, r_2, \cdots, r_m\}$ 为关系集合, 任意一条有向边表示一个三元组 (h, r, t) ($h, t \in E$ 和 $r \in R$). G_{kg} 也可看作三元组集合.

首先, 将给定的查询实体记为 e_q, 为了增加搜索候选实体的效率, 本文提出从全局重要度和局部重要度两方面来度量关系 r 对 e_q 的语义表示能力, 去除对 e_q 语义表示能力弱的关系, 减少需计算的关系数量.

(1) 全局重要度, 即关系 r 在 KG 中的重要程度. r 在 G_{kg} 中出现的频率越高, 其对 e_q 的特殊性就越小, 重要性也就越小. 按以下方式计算 r 对 e_q 的全局重要度:

$$I_1(e_q, r) = \frac{1}{r'} \tag{1}$$

其中, r'为 r 在 G_{kg} 中出现的次数.

(2) 局部重要度, 即关系 r 在以查询实体 e_q 为中心的局部子图中的重要程度. 将 KG 中与 e_q 直接相连的边构成的集合记为 $R'(e_q)$, r 在 $R'(e_q)$ 中出现的次数越多, 说明 e_q 通过 r 连接的实体越多, 进而 r 对 e_q 就越

重要. r 在 $R'(e_q)$ 中出现的次数与其重要程度成反比, 计算公式如下:

$$I_2(e_q,r)=\frac{r''}{|R'(e_q)|} \tag{2}$$

其中, r''为关系 r 在 $R'(e_q)$ 中出现的次数, $|R'(e_q)|$为 $R'(e_q)$ 中三元组的个数.

然后, 使用超参数 α 来平衡上述因素对关系 r 语义表示能力的贡献. 为了统一 $I_1(e_q, r)$ 和 $I_2(e_q, r)$ 的取值区间, 使用最大最小归一化函数 (Min-Max Scaling)[13] 对全局重要度和局部重要度进行处理, 计算公式如下:

$$I(e_q,r)=\alpha Nor(I_1(e_q,r))+(1-\alpha)Nor(I_2(e_q,r)) \tag{3}$$

其中, $\alpha\in[0,1]$, 为衡量各因素贡献比重的超参数, $Nor(\cdot)$ 为最大最小归一化函数.

最后, 为了提高候选实体搜索的效率, 通过式 (3) 计算 KG 中各关系对查询实体 e_q 的语义表示能力并对各关系进行排序, 选择其中得分最高的前 k 个关系, 记为集合 S.

2.2 查询实体关系选择

首先, 将 KG 中的实体通过训练嵌入到向量空间中, 得到对应的实体向量集 $E=\{e_1, e_2,\cdots, e_n\}$, 其中, $e_j\in E$ $(1\le j\le n)$ 是实体 e_j 的向量表示. 将与关系集合 S 对应的超平面法向量集记为 $D=\{d_1, d_2,\cdots, d_k\}$, 将与集合 D 中第 i 个法向量对应的关系记为 $r_i\in R(1\le i\le k)$. 使用式 (4) 计算实体 e_j 在 r_i 对应超平面上的投影, 如图 1 所示.

$$e_j^i=e_j-d_i^{\mathrm{T}}e_jd_i \tag{4}$$

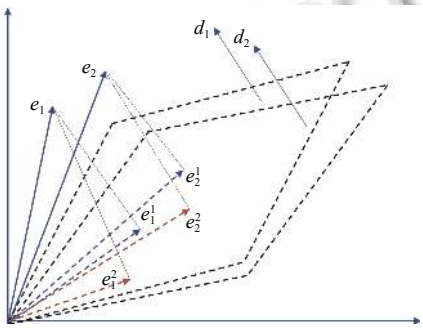

图 1 实体在各超平面中的投影

然后, 为了正确地在各超平面中搜索候选实体, 将每一个实体向量 e_j 在超平面 d_i $(1\le i\le k)$ 上的投影作为该实体在 r_i 对应超平面中的语义表示, 并根据实体在不同超平面中的语义表示, 将具有共同语义特征的实体划分为一类. 具体而言, 由于 K-means++算法[14] 的效率高、能够高效地对海量实体进行划分[15], 因此, 通过投影向量间的余弦相似度表示对应实体在 r_i 下的语义相似度, 使用 K-means++对 D 中各超平面上的实体投影进行聚类, 将与e_q^i同属一类的投影所对应的实体作为 d_i 上与 e_q 有共同语义特征的实体. 选择每个超平面中都与 e_q 同属一类的实体, 作为候选实体搜索的结果, 计算公式如下:

$$M(e_q)=M_1\cap M_2\cap\cdots\cap M_i\cap\cdots\cap M_k \tag{5}$$

其中, $M(e_q)$ 表示候选实体搜索的结果.

算法 1. 候选实体搜索算法

输入: e_q: 给定的查询实体; G_{kg}: KG; S: 对 e_q 影响最大的前 k 种关系集合

输出: $M(e_q)$: 候选实体集

1. 使用 TransH 将 G_{kg} 嵌入到向量空间中, 获得实体向量集 E 和与 S 对应的超平面法向量集 $D=\{d_1, d_2,\cdots, d_k\}$
2. for i=1 to k do
3. $N_i\leftarrow\varnothing$
4. for each e_j in E do
5. $e_j^i=e_j-d_i^{\perp}e_jd_i$ // e_j 在第 i 个超平面的投影
6. 将e_j^i添加到集合 U_i 中, 将第 i 个超平面中e_j^i与 e_j 的映射关系添加到 N_i 中
7. end for
8. K-means++(U_i) //对 U_i 中的实体投影进行聚类
9. 找到聚类结果中与e_j^i同属一类的投影在 N_i 中对应的实体, 将实体添加到集合 M_i 中
10. end for
11. $M(e_q)=M_1\cap M_2\cap\cdots\cap M_i\cap\cdots\cap M_k$
12. return $M(e_q)$

算法 1 主要的时间代价是在 k 个超平面中对实体投影进行聚类, 假设聚类类别数为 n', 每一次聚类的时间复杂度为 $O(n'n)$[14], 因此, 算法 1 的时间复杂度为 $O(kn'n)$.

3 相关实体排序模型

3.1 EUWG 模型

将需构造的无向带权图记为 G_{eg}, V 是 G_{eg} 中的节点, 由查询实体 e_q 与候选实体组成, 使用 V'表示 V 中除 e_q 外的实体集合. 由于查询实体与各候选实体相关, 因此, 先在 G_{eg} 中构造 e_q 与 V'各实体间的无向边, 然后, 通过计算各候选实体对应向量间的余弦相似度来构建候选实体间的无向边. 将 V'中任意两实体记为 v_i 和 v_j, 若 v_i 和 v_j 对应向量间的余弦相似度为正, 则在

G_{eg} 中构造一条 v_i 到 v_j 的无向边, 表示 v_i 与 v_j 相关. 下面给出 EUWG 模型的定义:

定义 2. EUWG 模型是一个无向带权图, 表示为 $G_{eg}=(V, L, M)$, 其中, $V=M(e_q)\cup\{e_q\}$为节点集合, $L=\{l_1, l_2,\cdots, l_t\}$为边的集合, $W=\{w(v_i, v_j)|1\le i, j\le s, v_i, v_j\in V, i\ne j\}$为 EUWG 边上的权重集合, $w(v_i, v_j)$ 表示 v_i 和 v_j 间无向边上的权重.

为了计算 G_{eg} 中边上的权重, 并描述节点间的相关程度, 我们考虑以下两个方面:

(1) 向量相关度. 各实体在向量空间中的语义相关度决定其向量间的相关度, 使用实体向量间的余弦相似度来度量. 余弦相似度越高, 结构相关度越大. 计算方法如下:

$$y_1\left(v_i, v_j\right)=Sim\left(v_i, v_j\right) \tag{6}$$

其中, $Sim(\cdot)$ 表示实体向量间的余弦相似度.

(2) Web 文档相关度, 即 G_{eg} 中任意两个节点在 Web 文档中共现频率反映的相关度[3]. 我们统计G_{eg}中任意两个节点在 Web 文档中共同出现的次数, 次数越多, 相关度越大. 将 Web 文档集合记为 $H=(h_1, h_2,\cdots, h_c)$, 计算方法如下:

$$y_2\left(v_i, v_j\right)=\sum_{x=1}^{c} g(h_x, v_i, v_j) \tag{7}$$

其中, 若实体 v_i 与 v_j 共同出现在 h_x $(1\le x\le c)$ 中, 则 $g(h_x, v_i, v_j)$ 为 1, 否则为 0.

使用超参数 β 来平衡上述因素对 G_{eg} 边上权重的贡献. 为了统一 $y_1(v_i, v_j)$ 和 $y_2(v_i, v_j)$ 的取值区间, 使用最大最小归一化函数对其进行处理:

$$w\left(v_i, v_j\right)=\beta Nor\left(y_1\left(v_i, v_j\right)\right)+(1-\beta)Nor\left(y_2\left(v_i, v_j\right)\right) \tag{8}$$

3.2 候选实体打分排序

G_{eg} 中任意两个节点间有多条路径, 不同的路径决定了节点间不同的相关程度. 因此, 通过获取查询实体 e_q 与候选实体 $v_i\in V'$在 G_{eg} 中的所有路径来计算每条路径上权重的平均值, 将其中的最大值作为候选实体 v_i 的分数, 并基于该分数对候选实体进行排序, 计算方法如下:

$$Score(e_i)=\max\left(\frac{\sum_{a=1}^{|z_j|-1} w\left(z_j^a, z_j^{a+1}\right)}{|z_j|}\right),\ j=1,2,\cdots,|Z_i| \tag{9}$$

其中, Z_i 表示查询实体 e_q 到候选实体 v_i 所有的路径集合, z_j 表示第 j 条路径需要经历的所有实体集合, z_j^a表示第 j 条路径中的第 a 个实体.

算法 2. 基于 EUWG 模型的候选实体排序

输入: e_q: 给定的查询实体; V: 候选实体集 $M(e_q)$ 与查询实体 e_q 的并集; L: G_{eg} 中边的集合
输出: B: 实体排序结果

```
1. i←1, j←1, tmp_B←Ø, B←Ø
2. for each v in V–{e_q} do
3.    Z←BFS(L, e, e_q) //使用广度优先算法获取 G_eg 中实体 e_q 到 v 的所有路径
4.    score←0
5.    for each z in Z do
6.    weight←0
7.       for a=0 to |z|–1 do
8.          weight←weight+w(z^a, z^(a+1))
9.       end for
10.      if weight/|z|>score then //将各路径权重平均值的最大值作为候选实体分数
11.         score←weight/|z|
12.   end for
13.   tmp_B←tmp_B∪{(v, score)} //tmp_B 保存候选实体 v 及其分数 score 组成的二元组 (v, score)
14. end for
15. 根据 tmp_B 中候选实体的分数, 对实体进行排序, 将排序结果保存在 B 中
16. return B
```

在算法 2 中, 假设 G_{eg} 的节点数为 s, 算法主要的时间代价是对 $s-1$ 个候选实体进行广度优先搜索, G_{eg}采用邻接矩阵存储, 每一次搜索的时间复杂度为 $O(s^2)$. 因此, 算法 2 的时间复杂度为 $O(s^3)$.

4 实验结果

4.1 实验设置

(1) 数据集与测试环境

为了测试本文提出方法的效果, 使用 Wikidata (http://dumps.wikimedia.org/wikidatawiki/entities) 作为测试数据集, 并从 Wikidata 中分别随机抽取部分三元组, 记为 KB50K 和 KB500K, 统计信息如表 1 所示. 使用 KORE[16] 与 ERT[17] 数据集作为验证数据集, 这两个数据集均使用人工处理的方法给出了涉及 IT、明星、游戏、电视剧、音乐与电影领域的多组查询实体与候选实体间的相关度, 统计信息如表 2 所示. 同时, 为了构造 EUWG 模型, 分别使用 KORE 与 ERT 数据集中各领域的查询实体作为关键词搜索 Web 文档, 统计信息如表 3 所示.

表 1 测试数据集

数据集	实体数量	关系数	三元组数量
Wikidata (2015年版)	18 555 814	1719	44 913 641
KB50K	51 935	192	580 921
KB500K	480 199	263	3891 246

表 2 验证数据集

数据集	查询实体数	相关实体数
KORE	21	420
ERT	40	937

表 3 Web 文档数据集

领域	IT	明星	游戏	电视剧	音乐	电影	总数
数量	4963	6576	6066	5705	10 896	10 883	45 089

实验使用 E5-2650v3 2.3 GHz 处理器, 2080Ti GPU, 128 GB 内存, 用 Python 作为编程语言, 并使用 Spark 和 TensorFlow 框架作为编程框架.

(2) 测试指标

使用准确率 (Precision, P)、召回率 (Recall, R) 以及 $F1$ 分值来评价算法 1 的有效性, 计算方法如下:

$$P=\frac{TP}{TP+FP},\quad R=\frac{TP}{TP+FN},\quad F1=\frac{2\times P\times R}{P+R} \tag{10}$$

其中, TP 为被正确搜索到的候选实体数, FP 为被错误搜索到的候选实体数, FN 为未被搜索到的候选实体数.

为了验证 EUWG 模型的有效性, 使用皮尔逊相关系数 (Pearson Correlation Coefficient, PCC)、斯皮尔曼等级相关系数 (Spearman Correlation Rank Coefficient, $SCRC$) 以及调和均值 (Harmonic Mean, HM) 来评价排序结果. 其中 PCC 表示测试结果与验证数据集中相关度分数的一致性, $SCRC$ 表示测试结果与验证数据集实体排序的一致性, HM 表示测试结果与验证数据集之间的综合一致性. 计算方法如下:

$$PCC=\frac{\sum\left(X-\frac{X}{AC}\right)\left(Y-\frac{Y}{AC}\right)}{\sqrt{\sum\left(X-\frac{X}{AC}\right)^2}\sqrt{\sum\left(Y-\frac{Y}{AC}\right)^2}} \tag{11}$$

$$SCRC=1-\frac{6\sum_{i=1}^{AC}b_i^2}{AC\left(AC^2-1\right)} \tag{12}$$

$$HM=\frac{2\times PCC\times SCRC}{PCC+SCRC} \tag{13}$$

其中, X 为测试结果中的候选实体分数集, Y 为验证数据集中各候选实体的分数集, AC 为候选实体数, b_i 为第 i 个实体在测试结果中的位置与验证数据集中位置的差值, PCC、$SCRC$ 和 HM 的值越接近 1, 说明结果越好.

4.2 候选实体搜索有效性测试

为了测试实体数量对算法 1 的影响, 分别在 KORE 与 ERT 上测试了候选实体搜索的准确率、召回率和 $F1$ 值, 如图 2 所示. 可以看出, 随着实体数量的增加, 各项指标都有所下降. 当实体数量从 1×10^5 增加到 5×10^5 时, 实体数量增加了 5 倍, 但召回率仅降低了 10%. 原因在于, 实体数量的增加使得 TransH 的学习结果更加准确, 并能够更有效地表示实体的语义, 进而使算法 1 在大规模的 KG 上也表现优异.

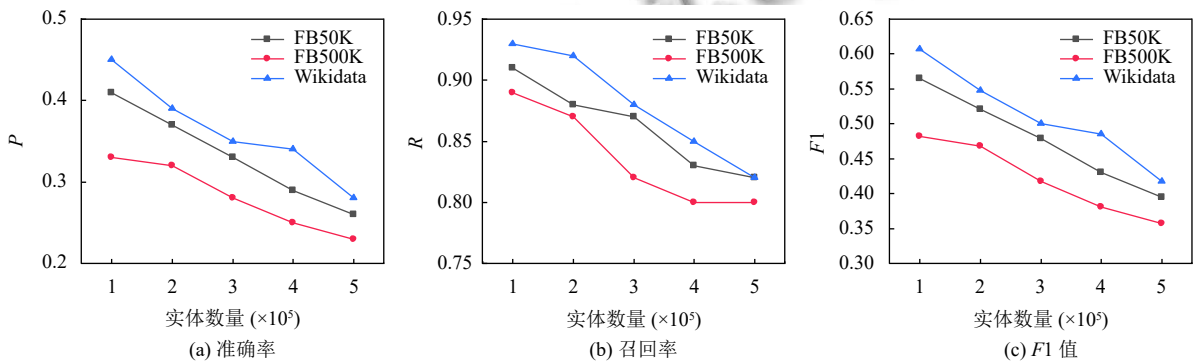

图 2 实体数对候选实体搜索的影响

然后, 测试了不同聚类类别数对算法 1 的影响. 在各 KG 中选择 5×10^5 个实体, 取不同的聚类类别数进行测试, 如图 3 所示. 可以看出, 随着聚类数的增加, 准确率和 $F1$ 值都有所上升, 原因在于类别数越多, 候选实体集中被错误召回的实体数量所占的比例越小, 进而候选实体搜索的准确性就越高.

另外, 将本文提出的候选实体搜索算法记为 TCES (TransH-based Candidate Entity Search), 从各 KG 中选

择 5×10^5 个实体, 设置聚类类别数为 170, 与 REFH[4] 和 LTRC[5] 算法进行对比, 如表 4 所示. 可以看出, 算法 1 在 FB50K 和 FB500K 数据集上效果更好, 且在 Wikidata 上准确率和 *F*1 值也高于其他两种方法. 原因在于, 算法 1 从 KG 所有实体中寻找候选实体, 搜索范围更大, 进而被正确召回的实体数目更多.

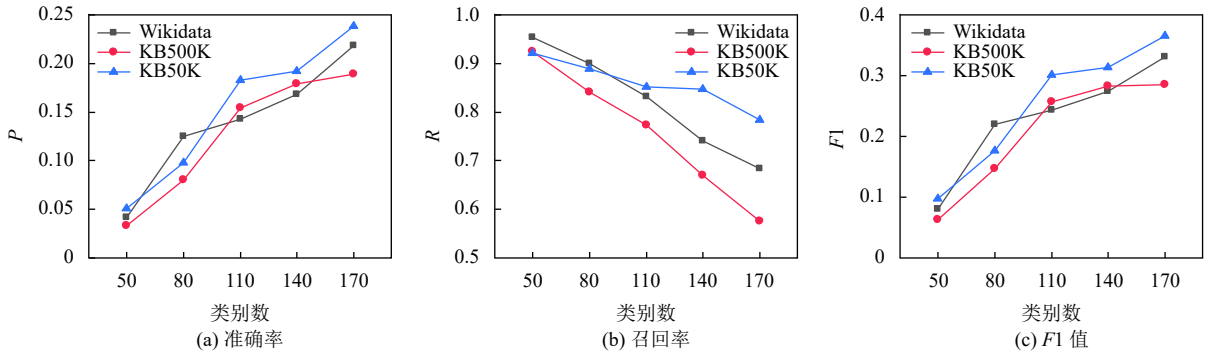

图 3 聚类类别数对候选实体搜索的影响

4.3 EUWG 模型有效性测试

为了测试 KG 规模对候选实体排序的影响, 选择 4.5×10^5 个 Web 文档, 测试算法 2 在不同三元组数量下的 *PCC*、*SCRC* 和 *HM*, 如图 4 所示. 可以看出, 随着三元组数量增加, *PCC*、*SCRC* 和 *HM* 都有所上升. 当三元组数量达到 5×10^6 时, 各指标平均增加了 29%、17% 和 25%. 原因在于随着三元组数量的增加, KG 中蕴含的知识更加完整, TransH 能够更有效地对 KG 进行表示, 使得 EUWG 模型中向量相关度的计算更加准确, 进而排序效果有所提升.

另外, 为了测试不同 Web 文档数对相关实体排序的影响, 从各 KG 中分别选择 5×10^6 个三元组, 测试算法 2 在不同 Web 文档数下的 *PCC*, *SCRC* 和 *HM*, 如图 5 所示. 可以看出, 随着 Web 文档数增加, 各指标也随之上升, 当数据量为 4.5×10^5 时, 各指标平均提升了 41%、30% 和 34%. 原因在于随着 Web 文档数的增加, 其中的知识也随之增加, 对实体相关性的描述信息也更加丰富, 使得 EUWG 模型对实体在 Web 文档中相关性的量化更加准确, 进而提升了排序效果.

表 4 不同 KG 的候选实体搜索结果

方法	FB50K			FB500K			Wikidata		
	P	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1
REFH	0.26	0.78	0.39	0.19	0.75	0.30	0.28	0.8	0.41
LTRC	0.21	0.71	0.32	0.22	0.64	0.32	0.30	0.69	0.42
TCES	**0.28**	**0.82**	**0.41**	**0.23**	**0.81**	**0.35**	**0.32**	0.77	**0.45**

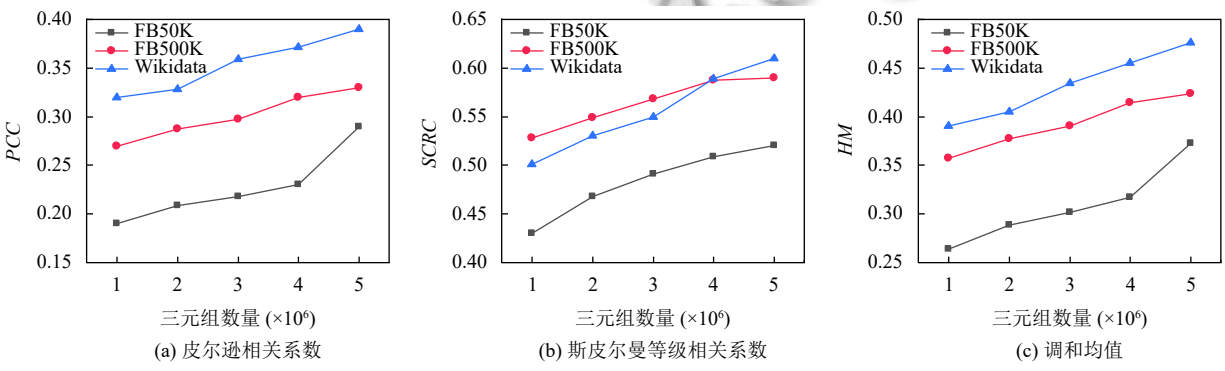

图 4 KG 规模对候选实体排序的影响

最后, 我们从 KG 中分别选择 5×10^6 个三元组, 并使用 4.5×10^5 个 Web 文档和不同领域的查询实体进行测试, 与 WLM[8]、TSF[9] 和 Wikipedia2Vec[12] 模型进行比较, 如图 6 和图 7 所示. 可以看出, 本文提出的 EUWG 模型在实体排序任务中表现较好, 其中, EUWG 模型比其他 3 种方法的 *PCC* 高了 18%. 原因在于 Wikipedia2Vec 模型在将 KG 映射为向量时会发生实体和词汇的匹配错误. 同时, WLM 与 TSF 模型主要根据 KG 来计算实

体间的相关度, 但 KG 无法及时地反映真实世界不断演化的知识, 因此计算结果不够准确, 而 EUWG 使用 Web 文档和 KG 共同描述实体间的相关关系, 使得计算结果更加客观, 进而候选实体的排序结果更好.

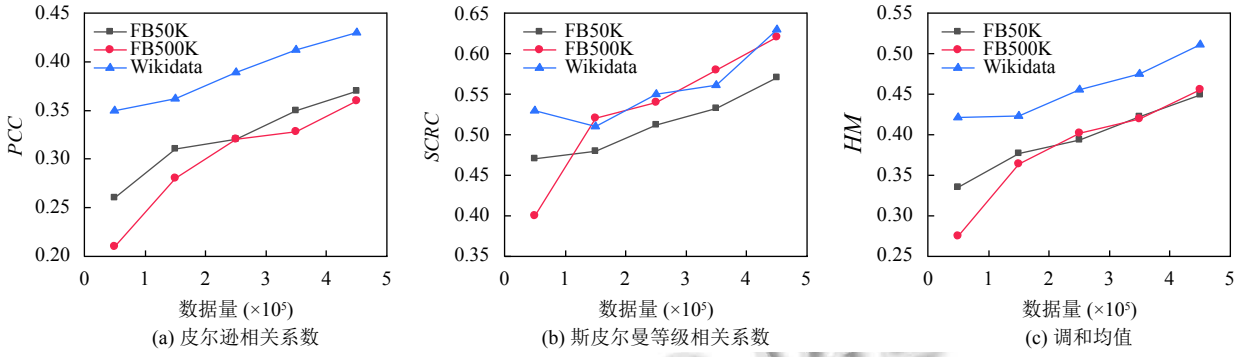

图 5 Web 文档数对候选实体排序的影响

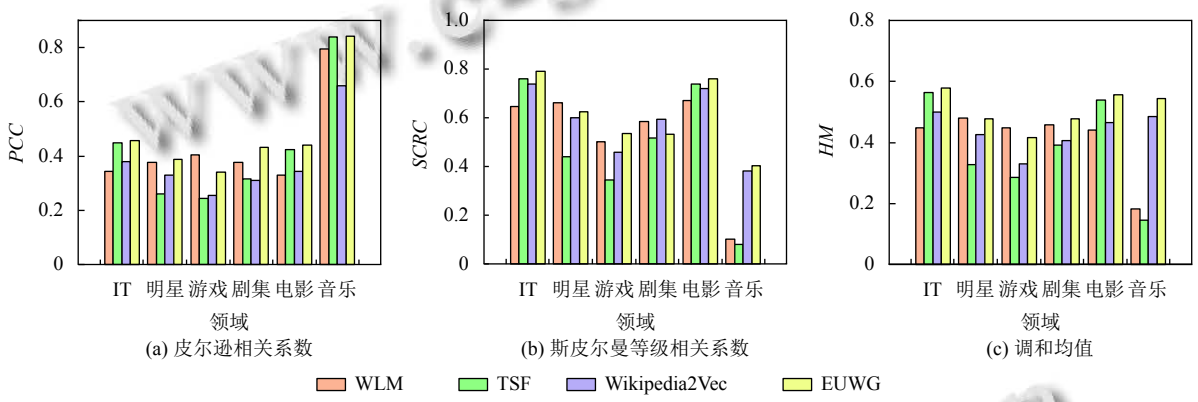

图 6 基于 FB50K 的候选实体排序结果

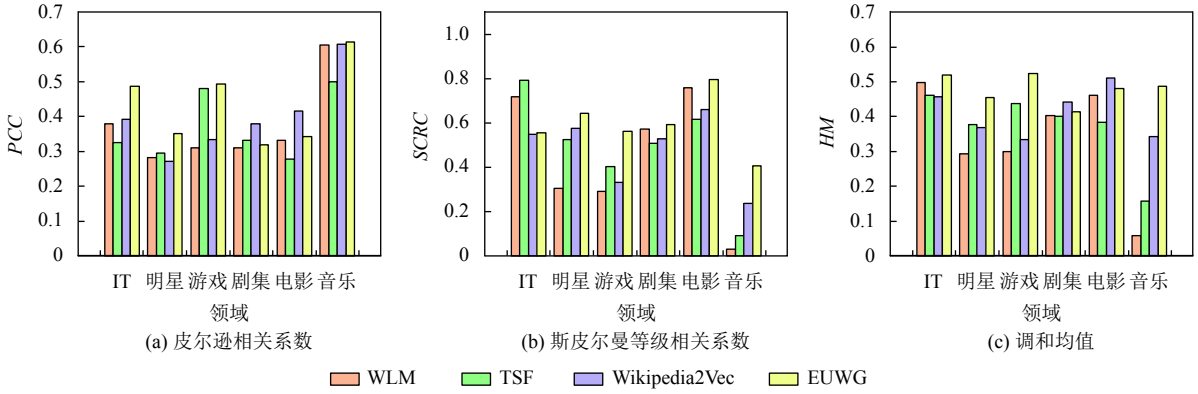

图 7 基于 FB500K 的候选实体排序结果

5 结束语

本文提出了基于表示学习的相关实体搜索算法, 通过对向量空间中不同关系超平面上的实体投影进行聚类, 获得与查询实体相关的候选实体, 并使用实体带权无向图模型对候选实体进行排序. 实验结果表明, 本文提出的方法能够正确地从 KG 中搜索候选实体, 同时有效地对候选实体进行排序. 但在候选实体排序任务中使用的数据源仍有待进一步扩展. 因此, 在未来工作中考虑加

入 Web 应用中与实体相关的图片数据, 更加客观全面地描述实体间的关系信息, 提高相关实体搜索的准确性.

参考文献

1 黄恒琪, 于娟, 廖晓, 等. 知识图谱研究综述. 计算机系统应用, 2019, 28(6): 1–12. [doi: 10.15888/j.cnki.csa.006915]

2 张香玲, 陈跃国, 马登豪, 等. 实体搜索综述. 软件学报, 2017, 28(6): 1584–1605. [doi: 10.13328/j.cnki.jos.005256]

3 Rahman MM, Takasu A, Demartini G. Representation learning for entity type ranking. Proceedings of the 35th Annual ACM Symposium on Applied Computing. New York: ACM, 2020. 2049–2056. [doi: 10.1145/3341105.3374029]

4 Reinanda E, Meij E, Pantony J, *et al*. Related entity finding on highly-heterogeneous knowledge graphs. Proceedings of 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. Barcelona: IEEE, 2018. 330–334. [doi: 10.1109/ASONAM.2018.8508650]

5 Huang JZ, Ding SQ, Wang HF, *et al*. Learning to recommend related entities with serendipity for web search users. ACM Transactions on Asian and Low-Resource Language Information Processing, 2018, 17(3): 25. [doi: 10.1145/3185663]

6 祁志卫, 王笳辉, 岳昆, 等. 图嵌入方法与应用: 研究综述. 电子学报, 2020, 48(4): 808–818. [doi: 10.3969/j.issn.0372-2112.2020.04.023]

7 Wang Z, Zhang JW, Feng JL, *et al*. Knowledge graph embedding by translating on hyperplanes. Proceedings of the 28th AAAI Conference on Artificial Intelligence. Quebec City: ACM, 2014. 1112–1119.

8 Milne D, Witten IH. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. Proceedings of 2008 AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy. Chicago: AAAI, 2008. 25–30.

9 Ponza M, Ferragina P, Chakrabarti S. On computing entity relatedness in wikipedia, with applications. Knowledge-Based Systems, 2020, 188: 105051. [doi: 10.1016/j.knosys.2019.105051]

10 Rothe S, Schütze H. CoSimRank: A flexible & efficient graph-theoretic similarity measure. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore: Association for Computational Linguistics, 2014. 1392–1402. [doi: 10.3115/v1/P14-1131]

11 Qian JW, Li XY, Zhang CH, *et al*. Social network de-anonymization and privacy inference with knowledge graph model. IEEE Transactions on Dependable and Secure Computing, 2019, 16(4): 679–692. [doi: 10.1109/TDSC.2017.2697854]

12 Yamada I, Shindo H, Takeda H, *et al*. Joint learning of the embedding of words and entities for named entity disambiguation. Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning. Berlin: Association for Computational Linguistics, 2016. 250–259. [doi: 10.18653/v1/K16-1025]

13 Wikipedia. Feature scaling. https://en.wikipedia.org/wiki/Feature_scaling. [2021-07-02].

14 Arthur D, Vassilvitskii S. K-means++: The advantages of careful seeding. Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms. New Orleans: ACM, 2007. 1027–1035.

15 LeLand M, John H. Benchmarking performance and scaling of Python clustering algorithms. https://hdbscan.readthedocs.io/en/latest/performance_and_scalability.html. [2021-08-30].

16 Hoffart J, Seufert S, Nguyen DB, *et al*. KORE: Keyphrase overlap relatedness for entity disambiguation. Proceedings of the 21st ACM International Conference on Information and Knowledge Management. Maui: ACM, 2012. 545–554. [doi: 10.1145/2396761.2396832]

17 Herrera JET, Casanova MA, Nunes BP, *et al*. An entity relatedness test dataset. Proceedings of the 16th International Semantic Web Conference on the Semantic Web. Vienna: Springer, 2017. 193–201. [doi: 10.1007/978-3-319-68204-4_20]