グラフ構造を利用したエンティティ検索

駒水　孝裕†

† 名古屋大学　〒 464-8601 愛知県名古屋市千種区不老町
E-mail: †taka-coma@acm.org

あらまし　Linked Data (LD) は，実世界における事実を記録し，Web 上で共有する枠組みである．エンティティ検索 (Entity Search) は LD を活用する際に重要なタスクのひとつである．エンティティ検索は，与えられたキーワードにマッチするエンティティを LD から検索するタスクである．従来のエンティティ検索の手法は，エンティティに対応する文書を構築し，情報検索や自然言語処理技術を用いて検索する．従来手法の文書構築は経験則に基づいており，汎用性の面で課題がある．加えて，従来手法のランキング性能は未だ改善の余地がある．本研究では，文書構築およびランキングにグラフ構造を利用した手法を適用することで，従来手法より検索性能を向上することを目的とする．本研究は，LD がグラフ構造を有することに着目し，経験則によらないエンティティの表現学習手法 RWRDoc を提案する．RWRDoc は Random walk with restart (RWR) をエンティティ間の距離尺度としてエンティティの表現学習をする手法である．RWRDoc は，エンティティまでのホップ数にかかわらず，距離の近いエンティティを関連するものとしてエンティティの表現を学習する．さらに，ランキング性能向上のために，グラフベースの再ランキング手法 PPRSD を提案する．PPRSD は文書ベースの関連度に加えて，グラフ構造に基づく関連性を考慮することで，ランキングの性能を向上させる手法である．本稿では，RWRDoc と PPRSD を組み合わせることで，エンティティの検索精度が向上することを実験的に示す．

キーワード　Linked Data, エンティティの表現学習，グラフベースの再ランキング，Random walk with restart

1 はじめに

Linked Data (LD) [1] は，データを公開する方法論のひとつで，異種ドメインのデータを関係性を表すリンクで結びつける．様々なオープンデータが LD として公開されており，代表的な LD データセットには，DBpedia[1] や統計 LOD[2] などがある．LD では，エンティティをノードとしエンティティ間の関係を関係性を表現するエッジで結ぶことで，ある事実を表現する．この事実が多く集まることで，知識の Web を構築する．

エンティティ検索 [17] は，与えられたキーワードクエリに対して，LD からクエリに適合するエンティティを見つけ出すタスクである．LD を活用する多くのアプリケーションでは，まず利用するエンティティを発見する必要がある．エンティティ検索の重要性から，これまでに様々なベンチマークが作成されてきた．例えば，INEX-LD [22], QALD [21], DBpedia-Entity [5] は代表的なベンチマークである．特に，DBpedia-Entity は近年よく利用されるベンチマークで，従来のベンチマークからクエリや検索結果を選別されて作られているため，幅広い問合せ要求をカバーしている．

エンティティ検索の従来研究は，情報検索や自然言語処理の技術をもとにしている．詳しいリストは DBpedia-Entity のホームページ [3] に記載されている．従来研究は，フィールド付文書モデル (Fielded document model) を採用している．このモデルは，エンティティをフィールド付きの文書として表現し，文書検索技術を適用する．エンティティ検索において，フィールドはエンティティを主語として持つトリプルの述語である．従来研究は，取りうる述語のうち，DBpedia 上での頻度上位 1000 件の述語をフィールドとして用いている．このフィールド付文書に対して，文書検索技術として，例えば，BM25 [19] を利用した手法や言語モデル [16] を利用した手法が提案されている．これらに加えて，フィールドごとの重要度を考慮するように拡張した手法も提案されている [13].

効果的なエンティティ検索を実現するためには，エンティティに対応する文書を適切に設計しなければならない．とくに，フィールドの設計はエンティティが検索にマッチするかどうかに関わる重要な項目である．本研究では，まず，従来研究が適切な文書を設計できているかを評価する．DBpedia-Entity では，検索性能の評価指標として NDCG (Normalized Discounted Cumulative Gain) [8] が用いられている．本研究では，検索結果の見落としを評価するために，再現率@k による評価を行う．2 節では，k を 10, 100, 1000 としたとき，従来研究は最大でも 0.2872, 0.6912, 0.8708 しか達成できていいないことを示す．これは，フィールドの設計やランキング手法に改善の余地が十分に残されていることを示している．

本研究では，LD が持つグラフ構造に着目し，エンティティの文書を学習する手法 RWRDoc を提案する．従来研究はフィールドをエンティティに隣接する述語と定義している．しかしながら，グラフ上の距離が離れたエンティティでも関係する

1：http://dbpedia.org/
2：http://data.e-stat.go.jp/lodw/
3：http://tiny.cc/dbpedia-entity

ものが存在する [10]．例えば，豊臣秀吉 [4] は戦国時代の武将で，文禄の役 [5] を行った人物である．豊臣秀吉と文禄の役は深く関係があるものの，豊臣秀吉のエンティティに隣接している述語（例えば，dbo:abstract）では文禄の役についてのテキストに到達できない．そのため，豊臣秀吉のエンティティに対応する文書には文禄の役についての情報は含まれない．文禄の役を含むためには，述語 dbo:subject で示される dbo:Japanese_invasions_of_Korea_(1592-98) に付随するテキストを取り入れる必要がある．このように，隣接しているノードだけでなく，グラフ上で離れた位置に存在するエンティティについても文書に取り入れる必要がある．グラフ上離れていても関係するエンティティを捕捉するために，Random walk with restart (RWR) [20], [24] によってエンティティ間の近接性を測り，近いエンティティに付随するテキストを文書に取り込む．RWRDoc では，RWR の近接性を重みとして，エンティティに付随するテキストから文書を学習する．

エンティティ検索の検索性能向上のため，エンティティの文書学習に加え，本研究ではグラフ構造に着目したエンティティのランキング手法 PPRSD を提案する．従来研究では，クエリに関連するエンティティのグラフ上の関係性を考慮していない．PageRank [15] の成功に代表されるように，グラフ構造を利用することは合理的である．しかしながら，PageRank をただ適用するだけでは，十分な検索性能を得られない [10]．そこで，本研究では，クエリのエンティティ文書に対する関連度をグラフ構造上で再分配することでグラフ構造を加味したエンティティのランキングを実現する．PPRSD (Personalized PageRank based Score Distribution) では，関連度を Personalized PageRank [6] の容量で再分配する．

本研究の実験で，RWRDoc および PPRSD が従来研究に対して，再現率@k および NDCG において改善していることを示す．RWRDoc はより広い範囲のテキストを文書として取り込むことで，関連するエンティティがクエリにマッチする可能性を向上させることが期待される．これを評価するために，再現率@k の観点で評価を行い，従来研究と比較評価する．PPRSD は計算済みの関連度をグラフ構造に従って再分配し再ランキングする．これにより，ランキング性能の向上が期待される．実験では，NDCG を用いて従来手法から PPRSD がどれだけ改善できたかを評価する．また，RWRDoc と PPRSD を組み合わせた場合とも比較評価を行う．実験では，エンティティ検索の最新ベンチマークである DBpedia-Entity [5] を用いる．

本研究の貢献を以下にまとめる．

- RWRDoc によるエンティティ文書学習手法．従来研究の問題点であるグラフ上で離れた距離にある関係するエンティティをエンティティ文書に含める手法 RWRDoc を提案する．RWRDoc では RWR を距離尺度として距離の近いエンティティに関するテキストをエンティティ文書に含めることでエンティティ文書学習を実現する．
- PPRSD によるエンティティランキング手法．エンティティ文書に対するクエリの関連度だけでなく，グラフ上でのエンティティの重要度を加味するランキング手法 PPRSD を提案する．PPRSD では，Personalized PageRank を模して関連度を再分配し，ランキング性能を改善する．
- RWRDoc+PPRSD によるエンティティ検索の精度向上．RWRDoc によるエンティティのマッチング性能向上と PPRSD によるランキング性能向上のふたつを組み合わせ，エンティティ検索の性能向上を実現する．実験により，RWRDoc と PPRSD を組み合わせることで，従来研究よりも高性能なエンティティ検索を達成することを示す．

本論文の構成は次のようになる．2 節では，エンティティ検索のタスクおよび従来手法の状況を説明する．3 節では提案手法である RWRDoc と PPRSD を紹介する．4 節で DBpeda-Entity [5] を用いた評価実験を示す．5 節で関連研究を紹介し，6 節で本論文の結論と今後の研究について述べる．

2 エンティティ検索

本研究を含むエンティティ検索手法では，LD に含まれる構造情報を用いる．そのため，LD データはデータグラフとして扱われる．データグラフの定義は定義 1 のようになる．

［定義 1］（データグラフ） ある LD データのデータグラフ $G = (V, E)$ は，ノード集合 V とエッジ集合 E からなるグラフである．ノード集合 $V = R \cup L \cup B$ は，エンティティの集合 R，リテラルの集合 L，ブランクノードの集合 B の和集合であり，エッジ集合 $E \subseteq V \times P \times V$ は，ノード間に張られるラベル付エッジの集合である．各エッジラベルは，LD データの述語の集合 P である．□

エンティティ検索は，入力キーワードにマッチするエンティティを LD データセットから検索するタスクである．エンティティ検索では，エンティティをどのように表現し，入力キーワードにマッチさせるかが中心的な課題である．エンティティ検索はこれまでに様々な研究がなされてきた．DBpedia-Entity ベンチマーク [5] では，従来研究の検索性能を記録している [6]．従来研究の基本的な手法としては，BM25 [19], BM25-CA [19], LM (Language Modeling) [16], SDM (Sequential Dependency Model) [11], PRMS (Probabilistic Model for Semistructured Data) [9], や MLM-all (Mixture of Language Models) [14] などの情報検索技術や自然言語処理技術を適用した手法がある．これらの手法に加え，フィールド重み付けを行う拡張を施した手法として，MLM-CA [14], FSDM (Fielded Sequential Dependence Model) [25], and BM25F-CA [19] がある．さらなる拡張として，クエリにエンティティリンキングを適用した，LM-ELR [4], SDM-ELR [4], FSDM-ELR [4] がある．

これらの従来手法は，エンティティをフィールド付き文書として表現している [3]．従来手法は，各エンティティを 1000

4：http://dbpedia.org/resource/Toyotomi_Hideyoshi

5：http://dbpedia.org/resource/Japanese_invasions_of_Korea_(1592-98)

6：https://github.com/iai-group/DBpedia-Entity

表 1: 再現率@k ($k = 10, 100, 1000$)．各列は再現率を計測したクエリセット (SemSearch ES, INEX-LD, ListSearch, QALD-2) を表し，最後の列 (Total) は全クエリセットのマクロ平均を表す．各列の部分列は k の値に対応し，最大の値は太字と下線で強調している．各行は手法に対応し，最後の二行は手法の最大値 (max) と@1000 のときに最大の再現率からの乖離 (gap) を表す．上位 1000 件では関連するエンティティを 80%以上検索できているにも関わらず，より上位にランキングできていないことがわかる．

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	—

フィールドに 3 フィールドを加えたフィールドを持つ文書として表現する．前者の 1000 フィールドは，DBpedia 上で最頻出する 1000 個の述語に対応する．後者の 3 フィールドは，特別に設計されたフィールドで，name, types, contents の三つである．name フィールドは，`rdfs:label` と `foaf:name` の述語の目的語になるテキストをすべて含むフィールドである．同様に，types フィールドは，`rdf:type` および接尾辞が “subject” である述語の目的語になるテキストをすべて含むフィールドである．最後に，contents フィールドは `owl:sameAs` 述語 以外[7] の述語で隣接するエンティティの 1000 フィールドのテキストをすべて含むフィールドである．上記手法は，これらのフィールドの一部を利用している．例えば，BM25, LM, SDM は contents フィールドのみを利用しているのに対し，MLM-all, PRMS, FSDM は最貧集する 10 フィールドを利用している．フィールド重み付け手法は，それぞれのフィールドに重みを与えて利用している．例えば，MLM-all はすべてのフィールドに対し一様の重みを与え，PRMS はフィールドの重みを学習する．

従来研究は近接したテキストを活用している手法であるため，関連するテキストを見落とす可能性がある．その結果として，検索性能が低下してしまう可能性がある．ここでは，従来研究の検索性能を評価し，エンティティ検索の性能向上のための手がかりを得る．そのために，ベンチマークで公開されている従来研究での検索結果 [8] を再現率@k (式 1) で再評価する．

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

表 1 に k が 10, 100, 1000 のときの再現率@k を示す．表では，各タスク (SemSearch ES, INEX-LD, ListSearch, QALD-2) と全タスクの平均 (Total) に分けて評価を行った．各行は従来研究に対応する．最後の二行は各タスクの各 k における最大の再現率 (max 行) と再現率@1000 からの乖離 (gap 行) を示している．各列について，最大の値を取るセルの値は太字と下線で強調している．この表から，上位 1000 件についての再現率はどのタスクも 80%を超えているのに対し，上位 10 件における再現率は 20%から 45%程度と低くなっていることがわかる．このことから，ランキング性能を向上させることで，それぞれの手法を改善することが示唆される．

3 RWRDoc + PPRSD

本研究では，LD が持つグラフ構造を利用し，エンティティ検索を効果的に行う手法を提案する．着眼点としては，(1) エンティティの文書を学習する際にグラフ構造を取り入れる，と (2) 検索結果のランキングにグラフ構造を取り入れる，の二点である．これらはいずれも従来研究で実現されていないため，本研究でグラフ構造を取り入れることの優位性を示す．それぞれの着眼点に対して，本研究では RWRDoc と PPRSD を提案する．以下の節ではそれぞれについて紹介する．

3.1 RWRDoc: RWR に基づくエンティティ文書学習

2 節で述べたように，従来研究はフィールドに隣接する述語のみを利用しているため，再現率が十分でないという問題があった．本研究ではこの問題に対し，LD 上で関連するエンティティを文書に取り入れる方法を提案する. [10] でも指摘されているように，あるエンティティから到達可能なエンティティの数はそのエンティティからの距離（ホップ数）を増やすごとに爆発的に増加し，関連性の薄いエンティティにも到達する可能性がある．そのため，ある距離で到達可能なエンティティを文書に取り入れる，というアイデアは現実できない．そこで，本研究では，ランダムウォークに基づく距離尺度でエンティティ間の関連性を測る．具体的には，Random walk with restart

7：`owl:sameAs` では同じエンティティの別言語バージョンに接続されていることが多く，同様の情報が重複することを避けるために除外されている．

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

(RWR) [20] を利用する．RWR は，あるグラフ上において，あるノードを始点としランダムサーファーが他のノードに到達する確率を計算するモデルである．RWR では，一定確率で始点のノードに戻るという操作 (with restart に相当) を加えることでより距離の近いノードへの到達確率が高くなる．これは本研究の目的に有効な性質である．あるエンティティから離れた位置にあるエンティティは関連性が薄く，近い位置にありよく到達されるエンティティは関連性が高いと考えられる．ノード u における他のノードに対する RWR 値は以下の式で表される．

$$\mathbf{rwr}_u = d \cdot \mathbf{rwr}_u \cdot A + (1-d) \cdot \mathbf{z} \qquad (2)$$

ただし，A はグラフの隣接行列を表し，$\mathbf{z}$ は u に対応する要素を 1 とし，他の要素を 0 とするワンホットベクトルである．d はダンピングファクターで，restart が行われる確率を表す．

LD データのデータグラフ G（定義 1）においては，A は G の誘導部分グラフ $G' = (R, E')$ から計算する．G' は G からエンティティをノードとするグラフを抜き出した誘導部分グラフ（定義 2）である．つまり，$E' = (R \times P \times R) \cap E$ である．誘導部分グラフを導入する理由は，LD データにおいてリテラルはいかなるトリプルの主語にもならないため，データグラフ上で末端ノードとなる．末端ノードが存在するグラフでは，RWR 値は末端ノードに吸い上げられてしまう．このため，末端ノードとなりうるリテラルを排斥している．

［定義 2］（誘導部分グラフ） データグラフ $G = (V, E)$ とノード集合 $V' \subseteq V$ に対して，誘導部分グラフ $G' = (V', E')$ は，V' をノード集合，G に存在するエッジ集合のうち V' のノード間に張られるエッジ集合 $E' = (V' \times V') \cap E$ をエッジ集合とする G の部分グラフである．□

RWRDoc は，RWR で計測した関連エンティティをエンティティの文書に取り込む．そのために，各エンティティを文書として取り入れられる形式にする必要がある．本研究ではこの形式を 最小エンティティ文書 と呼ぶ（定義 3）．

［定義 3］（最小エンティティ文書） エンティティ $v \in R$ の最小エンティティ文書 $\mathbf{m}_v$ は長さ $|W|$ のベクトルで表現される．ただし，W は単語の語彙集合とする．$\mathbf{m}_v$ の各要素は単語に対応し，単語の v への寄与度を表す．□

本研究では，TFIDF 値を寄与度とする [9]．最小エンティティ文書を計算するために，RWRDoc はまずエンティティに隣接するテキストを下記 SPARQL クエリを使って獲得する．

```
SELECT ?entity ?vals
WHERE { ?entity ?p ?vals.
        FILTER isLiteral(?vals). }
```

次に，獲得したテキストを bag of words の形式に変換し，TFIDF を以下の式で計算する．

9：各単語のエンティティへの寄与度は，任意に設定可能である．単語頻度や他の値でも RWRDoc での文書学習は可能である．

Algorithm 1 RWRDoc

Input: $G = (V, E)$: LD dataset
Output: $\mathbf{X}$: Learned Representation Matrix

```
Minimal Representation Matrix M, RWR Matrix Z
G' ← DataGraph(G)
for v ∈ R do
    M[v] ← TFIDF(v, G)
    Z[v] ← RWR(v, G')
end for
X = Z · M
```

$$\mathbf{m}_v = \Big(tf(t,v) \cdot idf(t,R) \Big)_{t \in W} \qquad (3)$$

$tf(t,v)$ はエンティティ v における t の出現頻度を表し，$idf(t,R)$ はエンティティ集合 R における t の逆エンティティ頻度を表す．

RWRDoc では，エンティティ $u \in R$ の文書表現 $\mathbf{x}_u$ を，すべてのエンティティの最小エンティティ文書について RWR を重みとした線形結合で計算する．

$$\mathbf{x}_u = \sum_{v \in R} rwr_{u,v} \cdot \mathbf{m}_v \qquad (4)$$

ただし，$rwr_{u,v} \in \mathbf{rwr}_u$ は u の v に対する RWR 値を表す．表現の単純化のために，$\mathbf{M}$ をすべてのエンティティの最小エンティティ文書を並べた行列とする．これにより，上記式を $\mathbf{x}_u = \mathbf{rwr}_u \cdot \mathbf{M}$ のように変形する．更に，$\mathbf{Z}$ を各エンティティごとの RWR ベクトルを集めた行列とすると，すべてのエンティティ文書を $\mathbf{Z}$ と $\mathbf{M}$ の積で表現できる．つまり，すべてのエンティティ文書を表す行列 $\mathbf{X}$ は $\mathbf{X} = \mathbf{Z} \cdot \mathbf{M}$ で計算できる．

Algorithm 1 に LD データ G からエンティティ文書を計算するアルゴリズム RWRDoc の手順を示す．まず，二行目では RWR を計算するための誘導部分グラフを導出する．三行目から六行目にかけて，それぞれのエンティティについて最小エンティティ文書と RWR を計算する．それぞれの計算結果を行列 $\mathbf{M}$ と $\mathbf{Z}$ に格納する．最後に，それらの積を計算し，すべてのエンティティについてのエンティティ文書 $\mathbf{X}$ を得る．RWRDoc の実装には，scikit-learn の TFIDF vectorizer[10] と RWR を高速に計算する TPA algorithm [24] を用いた．

3.2 PPRSD: グラフ構造を利用した再ランキング

従来の研究とはことなり，本研究ではエンティティ検索のランキングにグラフ分析の観点を取り入れる．従来の研究では，エンティティを表現する文書を作成し，文書に対するクエリの適合度合いを計算することで検索結果を計算していた．しかしながら，LD はグラフ構造を持つデータであり，グラフ分析的な手法を取り入れるのは理にかなっている．本研究では，グラフ分析的な手法として代表的な PageRank [15] に着目する．極めて自明であるが，単純にクエリに含まれる単語を有するエンティティを LD 上での PageRank 値でランキングすると，クエリの内容を反映しないため，非常に悪い結果となった [11]．

10：http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
11：予備実験を行い確認したが，結果が自明なためここでは結果は省略する．

提案手法 PPRSD では，クエリとの適合度合いを導入する方法として，Personalized PagerRank (PPR) [6] を導入する．PPR では，検索にマッチしたノードにのみランダムジャンプにより到達するように設計することで検索にマッチしたノードおよびその周辺のノードの PageRank 値を高くなるように計算する．クエリとエンティティの文書的な適合度合いを加味するために，PPRSD ではランダムジャンプする先の到達確率を適合度合いによって重み付けする．これは，従来のエンティティ検索に関する研究が合理的で精度のよい適合度合いを計算できており，これを利用するべき，というアイデアに基づいている．

PPRSD の計算手順は次のようになる．まず，与えられたクエリに対して，文書に基づく適合度を計算し，上位 1000 件を取得する．次に，取得した上位 1000 件のエンティティをノードとする誘導部分グラフを導出する．最後に，導出した誘導部分グラフ上で PPR に基づく関連度合いを計算し，関連度合い順にランキングする．PPRSD で上位 1000 件のエンティティの誘導部分グラフ（定義 2）を導入するのは，従来の手法でも 80%以上の再現率を上位 1000 件で達成できており（2 節参照），上位 1000 件からより関連するものを上位に押し上げる再ランキングの方が適切であると考えているためである．

PPRSD の関連度合いを計算（式 5）は，PPR の計算方法の一部を改変して行う．

$$\mathbf{pprsd}_q = (1-d)\cdot \mathbf{pprsd}_q A + d\cdot \mathbf{t} \qquad (5)$$

ここで，$\mathbf{pprsd}_q$ は長さ 1000 のベクトルで，各エンティティについての関連度合いを保持する．$\mathbf{s}$ は PPR での personalization vector で，文書的に関連するエンティティに対応する文書的な関連度を保持する長さ 1000 のベクトルである．従来手法の一部を PPRSD に適用する際に，$\mathbf{s}$ の設計で気をつける点がある．従来手法の一部には，対数尤度をスコアとしている手法がある (例えば，LM, MLM, SDM, FSDM, PRMS, やその拡張) ．これらの手法はスコアに負の値を取るため，PPRSD でスコアを再配分するのには向かない．そのため，予めスコアを指数関数を用いて正の値に変換する．更に，対数尤度を用いる手法は尤度関数の中身が確率の積になっており，指数関数で正の値にしても依然として小さい値（例えば，10^{-34}）になってしまう．これではほとんどスコアが考慮されなくなってしまうため，10^{-1} から 10^{-3} の範囲に収まるように定数倍・正規化を行う．式 5 は PageRank の計算で良く用いられるべき乗法で計算する．

4 エンティティ検索の精度評価

本実験では，提案手法 RWRDoc，PPRSD のエンティティ検索における精度を評価する．本実験では，以下の質問に答える．

- **質問 1：** グラフ上の距離を用いたエンティティの文書学習は再現率の向上に有用か？従来の隣接するテキストを用いる方式に対して，グラフ上の距離 (RWR) を用いて柔軟に学習した RWRDoc が再現率を向上したかを確認する．これにより，特定の述語ににによらず，エンティティ毎に必要なテキストがデータグラフ上に点在していることを示す．
- **質問 2：** グラフ分析はエンティティ検索のランキング性能向上に有効か？従来の文書館類似度を用いる手法に対して，データグラフの構造を反映した PPRSD が従来の手法を改善できたかを確認する．これにより，グラフ分析手法を取り入れる妥当性を示す．
- **質問 3：** それぞれを合わせた手法はエンティティ検索の精度向上に有効か？本研究の提案手法 RWRDoc と PPRSD を合わせることで，より高性能なエンティティ検索が実現できたかを確認する．これにより，エンティティ検索において LD のグラフ構造を利用する重要性を強調する．

上記質問に答えるために，提案手法と従来研究をエンティティ検索の精度の観点で比較する．計測する指標としては，再現率@k（式 1）と NDCG@k（式 7）で評価する．NDCG はランキングを評価する際に広く用いられる指標で，従来研究も NDCG で評価されているため，これに従う．　NDCG は理想的なランキングが与えられたときの DCG (Discounted Cumulative Gain, 式 6) に対する実際に計算したランクングの DCG の割合で計算される．DCG は関連する要素を上位に位置づけたときに値が高くなるように設計されいてる．DCG は要素の関連度合いを考慮に入れることが可能である（式 6 の rel_i）．しかし，今回のタスクにおいては，ランキング中のエンティティがマッチするかどうかの二択であるため，rel_i は 0（関連しない）か 1（関連する）のどちらかの値を取る．

$$DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{\log_2(i+1)} \quad (6) \qquad NDCG@k = \frac{DCG@k}{IDCG@k} \quad (7)$$

従来研究と比較するため，エンティティ検索において標準的に用いられるベンチマーク DBpedia-Entity v2 [5][12] を用いる．このベンチマークには，クエリとクエリにマッチするエンティティのリストが含まれる．クエリの種類としては，次の四種類が含まれる．(1) SemSearch ES：エンティティの隣接テキストにキーワードが含まれる単純なキーワード検索タスク；(2) INEX-LD：SemSearch ES より広い範囲の要求を含むキーワード検索タスク；(3) ListSearch：エンティティのリストを要求するキーワード検索タスク；(4) QALD-2：自然言語文による問合せタスク．このベンチマークの Web ページには，従来研究の NDCG@k のスコアおよび，検索結果の上位 1000 件が公開されている．PPRSD は後者のデータを利用して再ランキングを行う．ベンチマークで利用しているデータに従い，対象となる LD データは DBpedia 2015_10 データセット [13] である．

表 2 と表 3 に再現率@k と NDCG@k の結果を示す．各表の列は，ベンチマークのクエリセットに対応し (SemSearch ES, INEX-LD, ListSearch, QALD-2)，最右列はすべてのクエリセットのマクロ平均 (Total) に対応する．各列は k の値によって部分列に分かれている．各列において，既存手法で最大の値は 下線 で，全手法のうちで最大の値は**太字**で強調している．

表 2 は，上位 1000 件において RWRDoc が再現率の値の改善

12：https://github.com/iai-group/DBpedia-Entity
13：http://downloads.dbpedia.org/2015-10/

表 2: 再現率@k ($k = 10, 100, 1000$)．表 1 と同様の記載方法である．最下行は提案手法の RWRDoc と RWRDoc と PPRSD を組み合わせた手法 (RWRDoc*) である．RWRDoc と RWRDoc*の下の Imp. 行は既存手法からの改善率を既存手法の最大再現率に対する再現率の比で示す．各列について，既存手法で最大の値は下線で，すべての手法で最大の値は太字で強調している．RWRDoc は上位 1000 件の再現率を向上し，PPRSD と合わせることで，上位 10 件，100 件ともに改善していることがわかる．

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
RWRDoc	.4001	.8303	.9801	.2408	.6391	**.8624**	.2177	.5902	**.8613**	.2390	.6433	**.8298**	.2744	.6757	**.8834**
Imp. (%)	-11.79	-4.64	-0.65	-7.71	-2.07	+0.24	+3.18	+5.73	+2.16	-9.81	+1.45	+1.64	-2.51	-2.24	+1.45
RWRDoc*	.4325	.8511	.9801	**.2618**	**.6671**	**.8624**	**.2307**	**.6582**	**.8613**	**.2655**	**.6716**	**.8298**	**.2976**	**.7120**	**.8834**
Imp. (%)	-4.65	-2.25	-0.65	+0.42	+2.22	+0.24	+9.34	+5.13	+2.16	+0.19	+5.91	+1.64	+3.62	+3.01	+1.45

を示し，PPRSD と組み合わせる (RWRDoc*) ことで上位 10 件，上位 100 件においても再現率の改善を示している．RWRDoc は近距離のリテラルだけでなく，遠くのリテラルも加味する．これにより，再現率が改善してことがわかる．SemSearch ES は RWRDoc にとっては少し苦手なタスクであると言える．SemSearch ES は近傍のテキストに含まれるようなキーワードをクエリにしたタスクである，そのため，RWRDoc では近傍以外のテキストも「余分に」含んでしまって最良とはならなかった．しかしながら，一般的には他のタスクのように近傍以外のテキストを利用したクエリが存在する．また，全体的な再現率として最も優秀な値であるため，RWR を利用しエンティティ文書を学習する手法は適当であったといえる．さらに，PPRSD での再ランキングを取り入れることで，ランキングを修正し上位 10 件や上位 100 件でも総合的に最良の手法となった [14]．

表 3 は，PPRSD が既存手法のランキングを改善し，RWRDoc と組み合わせることで最良のランキング性能を達成したことを示す．表において，手法名に*がついたものは PPRSD での再ランキングであることを表し，Rise 行で PPRSD による改善率を示しており，改善率が負値の場合にセルを青色で強調している．これにより，ほぼすべての手法・タスクに置いて PPRSD がランキングを改善していることがわかる．加えて，PPRSD で悪化した手法・タスクの改善率は-0.1%以下と誤差の範囲であると言える．このことから，PPRSD によるランキングは既存のエンティティ検索のランキングを改善したことがいえる．表 2 で，RWRDoc により上位 1000 件の再現率が改善したことから，PPRSD により RWRDoc のランキングを再ランキングすることで，正答になるエンティティをより上位にランキングすることが期待される．Imp. 行により，RWRDoc* が既存手法に対して総合的に高いランキング性能を示している．しかし，SemSearch ES タスクや QALD-2 の上位 10 件のランキングでは，ランキング性能で既存手法を下回ってしまっている．前者は再現率と同じく，RWRDoc の性質によるものであると考えられる．後者は，RWR が LD 上のエッジの種類を無視しているために，意味的なつながりを無視してしまったために性能の十分な向上が達成できなかったと考えられる．

質問に対する回答は以下のようになる．

- **質問 1：** グラフ上の距離を用いたエンティティの文書学習は再現率の向上に有用か？
  回答： RWRDoc は上位 1000 件の再現率向上に強く寄与するが，上位 10 件，100 件の向上には有用ではなかった．これは，RWRDoc で構築したエンティティ文書により，クエリとのマッチングを見落としにくくする一方で，マッチング度合いについては不十分であったと言える．
- **質問 2：** グラフ分析はエンティティ検索のランキング性能向上に有効か？
  回答： ベンチマークを用いた実験により，PPRSD は既存手法において，ほぼすべてのクエリセットでランキング性能を向上した．一部ランキング性能が低下したものの，誤差の範囲と言えるほどの低下であった．故に，グラフ分析がランキング性能向上に有効であることが示唆された．
- **質問 3：** それぞれを合わせた手法はエンティティ検索の精度向上に有効か？
  回答： PPRSD のランキング性能向上は，既存手法に限らず RWRDoc でも改善が見られた．その結果，RWRDoc が苦手としていた上位のランキング（上位 10 件，100 件）においても既存手法を上回る性能を示した．故に，RWRDoc

14：既存手法についても PPRSD を適用した再現率実験を行っているがスペースの都合で省略している．興味がある方は [10] を参照されたい．

表 3: NDCG@k ($k = 10, 100$). 表 1 と同様に，各列はクエリセットと全クエリセットのマクロ平均を表す．各列の部分列は上位 10 件，100 件の結果に対応する (@10, @100)．各列では，既存手法で最大の値は下線で，すべての手法で最大の値は太字で強調している．各行は比較手法に対応し，*つきの手法は PPRSD で再ランキングを行ったものである．Rise 行は PPRSD による手法の改善率を表し，Rise が負値の場合にセルを青色で強調している．最下行の Imp. 行は，PPRSD での改善手法を含む既存手法のうち最大の NDCG@k からの RWRDoc*の改善率を示す．PPRSD で既存手法が改善され，RWRDoc のランキング性能問題を PPRSD が補い，ランキング性能が改善されていることがわかる．

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
Rise (%)	+13.7	+8.59	+58.8	+5.65	+304	+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
Rise (%)	+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
Rise (%)	+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
Rise (%)	+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
Rise (%)	+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
Rise (%)	+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
Rise (%)	+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
Rise (%)	+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
Rise (%)	+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
Rise (%)	+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
Rise (%)	+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
Rise (%)	+0.14	+0.69	0.00	+0.02	-0.02	0.00	-0.06	-0.02	-0.07	+0.15
RWRDoc	.5877	.7215	.4189	.5296	.4119	.5845	.3346	.5163	.4348	.5643
RWRDoc*	.6379	.7288	.4413	**.5462**	**.4355**	**.6015**	.3591	**.5623**	**.4684**	**.6097**
Rise (%)	+8.54	+1.01	+5.35	+3.13	+5.73	+2.91	+7.32	+8.91	+7.73	+8.05
Imp. (%)	-2.94	-0.99	-1.80	+2.36	+1.23	+16.1	-2.92	+20.4	+0.24	+9.25

と PPRSD を合わせた手法は総合的に見て，エンティティ検索の精度向上に有効であった．

5 関 連 研 究

既存のエンティティ検索手法は，2 節で述べたように情報検索，特に文書検索の技術と自然言語処理技術を応用したものが大半である．本研究はこれに対し，LD の持つグラフ構造を有効に活用する手法であり，従来研究とは異なるアプローチをとっている．

表現学習は自然言語処理の分野やグラフ解析の分野でも活発に研究されている．自然言語処理分野では，Word2Vec [12] を始めとする文字列の表現を与えられた巨大な文書集合から学習する試みがされ，大きな成功を収めている．一方で，グラフ解析の分野では，グラフ上のノードを低次元ベクトルで表現する研究が行われている．Node2Vec [2] を始めとするノードの表現

をグラフ内のエッジを利用して学習し，ノード分類やリンク予測などのタスクで成功を収めている．これらの発展形として，ノードに付随するテキストを利用してノードの表現学習をする研究も行われてきている [7], [18], [23].

本研究では，グラフ上で近接するノードのテキストをエンティティの表現に貢献するものとして取り入れるため，上記の表現学習とは異なるアプローチをとっている．本研究には，周辺の関連エンティティがエンティティを表現するための情報を部分的に保持しているという考え方が根底にある．一方で，上記表現学習は，周辺のエンティティに対する相対位置をベクトル空間上に射影する関数を学習する，というアイデアであるため，本研究とは学習する表現の意図が異なる．しかしながら，上記表現学習手法も有用な手法であるため，同様のアイデアを取り入れることは本研究の発展に貢献すると考えられる．この点については，今後の課題としたい．

6 おわりに

本稿では，グラフ構造に基づくエンティティ検索手法として，エンティティの表現学習手法 RWRDoc と検索結果の再ランキング手法 PPRSD を提案した．RWRDoc は Random walk with restart (RWR) を利用して，あるエンティティに関連性の高いエンティティを見つけ，関連エンティティを用いて表現学習を行う手法である．PPRSD は検索結果の関連性スコアを Personalized PageRank のアイデアで再配分し，再ランキングを行う手法である．従来手法では，グラフ上での距離が 2 ホップ以上離れた関連エンティティを表現に含めることができなかったのに対し，RWRDoc では，RWR による距離尺度を導入することにより，より柔軟に関連エンティティをエンティティの表現に含めることに成功した．PPRSD では，テキストに基づく関連度を LD 上のグラフ構造に着目して，再配分し，検索結果を再ランキングすることで，ランキング性能を向上させた．PPRSD は，RWRDoc だけでなく，従来手法のランキング性能を向上させた．全体的なエンティティ検索性能として，RWRDoc と PPRSD を合わせた手法が最も良い性能を示した．

謝　辞

本研究の一部は JSPS 科研費 JP18K18056 の助成を受けたものである．

文　献

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

[2] A. Grover and J. Leskovec. node2vec: Scalable Feature Learning for Networks. In *SIGKDD 2016*, pages 855–864, 2016.

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

[4] F. Hasibi, K. Balog, and S. E. Bratsberg. Exploiting Entity Linking in Queries for Entity Retrieval. In *ICTIR 2016*, pages 209–218, 2016.

[5] F. Hasibi, F. Nikolaev, C. Xiong, K. Balog, S. E. Bratsberg, A. Kotov, and J. Callan. DBpedia-Entity v2: A Test Collection for Entity Search. In *SIGIR 2017*, pages 1265–1268, 2017.

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

[7] H. Ito, T. Komamizu, T. Amagasa, and H. Kitagawa. Network-Word Embedding for Dynamic Text Attributed Networks. In *SCSN@ICSC 2018*, pages 334–339, 2018.

[8] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[9] J. Kim, X. Xue, and W. B. Croft. A Probabilistic Retrieval Model for Semistructured Data. In *ECIR 2009*, pages 228–239, 2009.

[10] T. Komamizu. Graph Analytical Re-ranking for Entity Search. In *EYRE@CIKM 2018*, 2018. (to appear).

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

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781, 2013.

[13] F. Nikolaev, A. Kotov, and N. Zhiltsov. Parameterized Fielded Term Dependence Models for Ad-hoc Entity Retrieval from Knowledge Graph. In *SIGIR 2016*, pages 435–444, 2016.

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

[15] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, November 1999.

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

[17] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc Object Retrieval in the Web of Data. In *WWW 2010*, pages 771–780, 2010.

[18] P. Ristoski and H. Paulheim. RDF2Vec: RDF Graph Embeddings for Data Mining. In *ISWC 2016*, pages 498–514, 2016.

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

[20] H. Tong, C. Faloutsos, and J. Pan. Random walk with restart: fast solutions and applications. *Knowl. Inf. Syst.*, 14(3):327–346, 2008.

[21] R. Usbeck, A. N. Ngomo, B. Haarmann, A. Krithara, M. Röder, and G. Napolitano. 7th Open Challenge on Question Answering over Linked Data (QALD-7). In *ESWC 2017*, pages 59–69, 2017.

[22] Q. Wang, J. Kamps, G. R. Camps, M. Marx, A. Schuth, M. Theobald, S. Gurajada, and A. Mishra. Overview of the INEX 2012 Linked Data Track. In *CLEF 2012 Evaluation Labs and Workshop*, 2012.

[23] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang. Network Representation Learning with Rich Text Information. In *IJCAI 2015*, pages 2111–2117, 2015.

[24] M. Yoon, J. Jung, and U. Kang. TPA: Two Phase Approximation for Random Walk with Restart. *CoRR*, abs/1708.02574, 2017.

[25] N. Zhiltsov, A. Kotov, and F. Nikolaev. Fielded Sequential Dependence Model for Ad-Hoc Entity Retrieval in the Web of Data. In *SIGIR 2015*, pages 253–262, 2015.