Learning Interpretable Entity Representation in Linked Data

Takahiro Komamizu

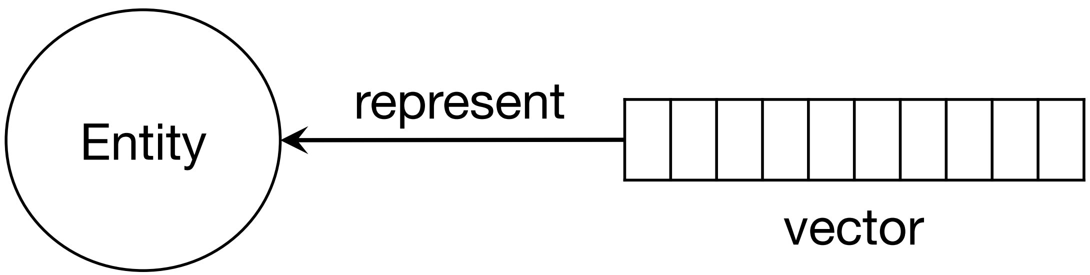
Nagoya University
Japan

Linked Data (LD)

- Open Data paradigm
- Consisting of simple factual descriptions
  - Triple: $(subject, predicate, object)$
    - $subject/object$: Entity (or literal for object)
    - $predicate$: Relationship
  - e.g., $(\langle Nagoya\_University \rangle, \langle located\_in \rangle, \langle Nagoya\_city \rangle)$
- Becoming a popular way of Open Data
  - e.g., LOD cloud (https://lod-cloud.net/, June 2018)
    - 1,220 datasets
      - Each dataset contains more than 1,000 triples.
    - 16,095 links between datasets

Entity Representation

- Feature design for entities in LD
- Originally, an entity is a node in a large graph.
- However, to deal with various tasks, entities should be represented as a **vector**.
    - Vector space model is a fundamental for many applications in data mining, information retrieval and so on.



Two Classes of Entity Representations

Interpretable

- Each element of vectors corresponds with interpretable thing (like terms in a document).
- e.g., TFIDF vectorization

Latent

- Each element of vectors has no clear meaning and is hard to interpret.
- e.g., Neural network-based methods

This paper prefers the **interpretable** representation.

- Interpretability is important to understand relationships b/w entities, like why they are similar.

Existing Interpretable Representations

Naive

Terms in literals connecting with entities

Predicate selection

Terms in literals connecting via **heuristically** selected predicates

Fielded Extension

Weighted terms with different weights for different predicates

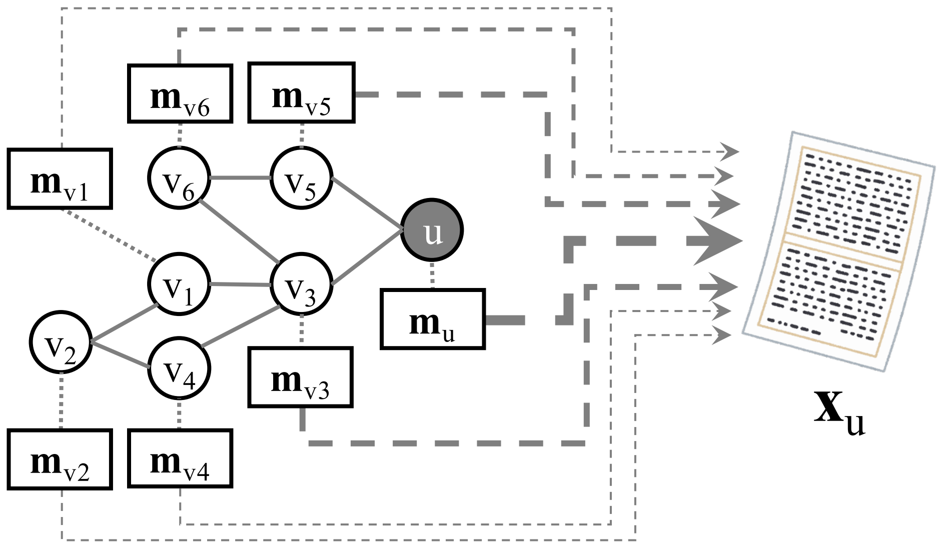
- Problems
  - How to select “good” predicates?
  - How can we design good weights for large variety of predicates?
  - Are the weights always same for different entities?

Research Objective

- Develop representation learning method which
  - representation is **interpretable**, and
  - **no heuristics** is required

RWRDoc: proposed approach

- Idea:
  - Entities “close” to the entity include relevant facts about the entity
- Approach: RWRDoc
  - TFIDF-based representation
  - Weighted sum of minimal rep.
  - Measuring closeness by <u>random walk with restart (RWR)</u>



Minimal Entity Representation

TFIDF vector for entity v

1. Obtain terms in surrounding literals

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

2. Calculate TFIDF values of terms

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

t is a term in vocabulary W
R is a set of all entities

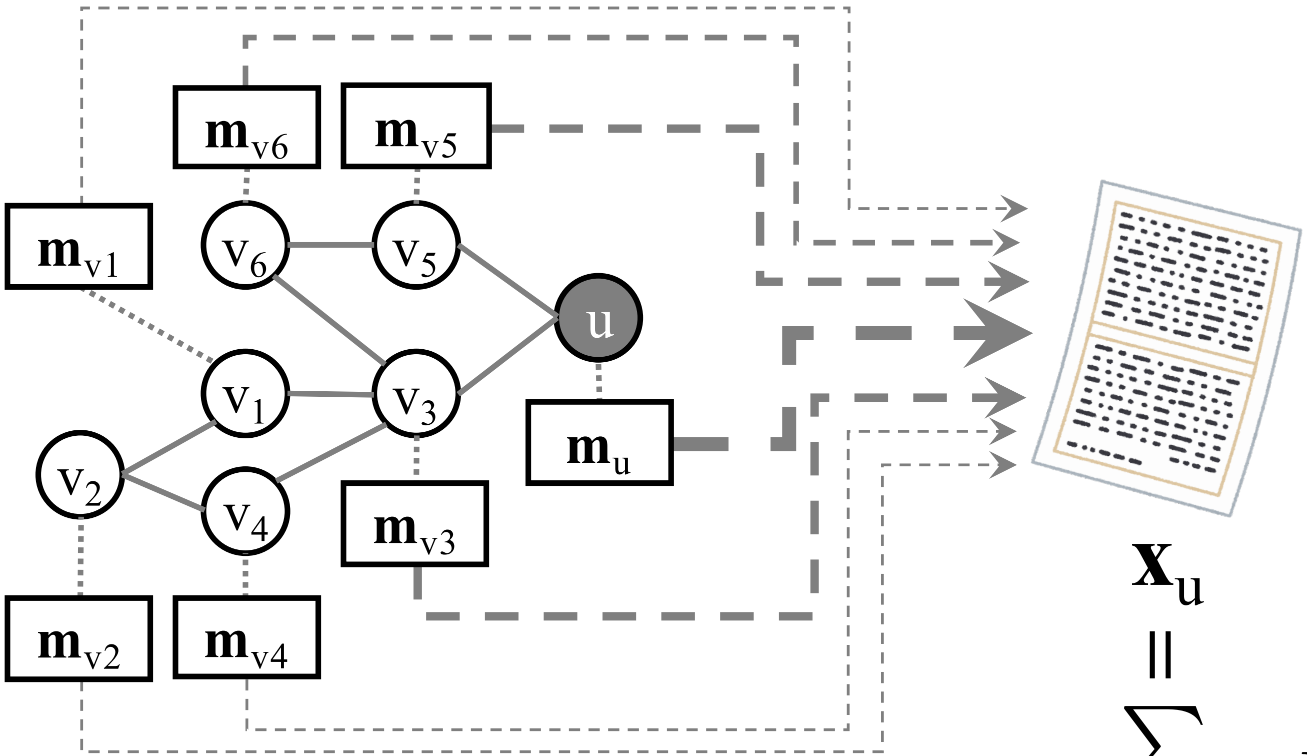
RWR: Random Walk with Restart

- A random surfer model on a graph
- Measuring probability random surfers arrive to nodes in the graph
- Restart: random surfers occasionally come back to the starting node and continue random walk

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

A is an adjacency matrix of the graph
$\mathbf{s}$ is a vector for restart which element for u is 1, 0 otherwise
d is damping factor

RWRDoc: minimal rep. × RWR



$$\mathbf{x}_u = \sum_{v \in R} z_{u,v} \cdot \mathbf{m}_v$$

RWRDoc: algorithm

Algorithm 1 RWRDoc

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

Minimal Representation Matrix $\mathbf{M}$, RWR Matrix $\mathbf{Z}$
$G' \leftarrow \mathit{DataGraph}(G)$ ▷ Prepare data graph G' for RWR computation.
for $v \in R$ **do**
    $\mathbf{M}[v] \leftarrow \mathit{TFIDF}(v, G)$ ▷ Calculate TFIDF vector for entity v.
    $\mathbf{Z}[v] \leftarrow \mathit{RWR}(v, G')$ ▷ Calculate RWR for source entity v.
end for
$\mathbf{X} = \mathbf{Z} \cdot \mathbf{M}$

- Implementation
    - TFIDF: scikit-learn TfidfVectorizer
    - RWR: TPA algorithm [26] (implemented by ourselves)
        - Quick approximation

Experimental Evaluation

Does RWRDoc learn good representation?

Generality

Applicability for various tasks

- direct use
- indirect use

Effectiveness

Qualities on various applications

Interpretability

Whether human judges can interpret entities

Tasks

- Entity search
- Recommender system with entity similarity
- Entity summarization

Entity Search Task

Given: LD datasets and a textual query (either keyword query or natural language query)
Find: Matching entities to the query from the datasets

- Benchmark: DBpedia-Entity v2 [8]
  - Quality measure: NDCG
- Input: a vector which elements corresponding with query terms are 1, 0 otherwise
- Similarity: cosine similarity

Ranking Quality on Entity Search

Easier tasks: SemSearch ES, INEX-LD. Harder tasks: ListSearch, QALD-2.

the state-of-the-art: BM25 through FSDM+ELR

Model	SemSearch ES		INEX-LD		ListSearch		QALD-2		Total	
top-k	@10	@100	@10	@100	@10	@100	@10	@100	@10	@100
BM25	0.2497	0.4110	0.1828	0.3612	0.0627	0.3302	0.2751	0.3366	0.2558	0.3582
PRMS	0.5340	0.6108	0.3590	0.4295	0.3684	0.4436	0.3151	0.4026	0.3905	0.4688
MLM-all	0.5528	0.6247	0.3752	0.4493	0.3712	0.4577	0.3249	0.4208	0.4021	0.4852
LM	0.5555	0.6475	0.3999	0.4745	0.3925	0.4723	0.3412	0.4338	0.4182	0.5036
SDM	0.5535	0.6672	0.4030	0.4911	0.3961	0.4900	0.3390	0.4274	0.4185	0.5143
LM+ELR	0.5554	0.6469	0.4040	0.4816	0.3992	0.4845	0.3491	0.4383	0.4230	0.5093
SDM+ELR	0.5548	0.6680	0.4104	0.4988	0.4123	0.4992	0.3446	0.4363	0.4261	0.5211
MLM-CA	0.6247	0.6854	0.4029	0.4796	0.4021	0.4786	0.3365	0.4301	0.4365	0.5143
BM25-CA	0.5858	0.6883	0.4120	0.5050	0.4220	0.5142	0.3566	0.4426	0.4399	0.5329
FSDM	0.6521	0.7220	0.4214	0.5043	0.4196	0.4952	0.3401	0.4358	0.4524	0.5342
BM25F-CA	0.6281	0.7200	**0.4394**	**0.5296**	**0.4252**	0.5106	**0.3689**	0.4614	**0.4605**	0.5505
FSDM+ELR	**0.6563**	**0.7257**	0.4354	0.5134	0.4220	0.4985	0.3468	0.4456	0.4590	0.5408
RWRDoc	0.5877	0.7215	0.4189	**0.5296**	0.4119	**0.5845**	0.3346	**0.5163**	0.4348	**0.5643**
Residual	-6.86%	-0.42%	-2.05%	0%	-1.33%	+7.03%	-3.43%	+5.49%	-2.57%	+1.38%

Score diff from the best/second best

Note that results for the state-of-the-arts are quoted from the benchmark paper [8]

Findings from Entity Search Task

Easier tasks (SemSearch ES, INEX-LD) — Harder tasks (ListSearch, QALD-2)

Model	SemSearch ES		INEX-LD		ListSearch		QALD-2		Total	
top-k	@10	@100	@10	@100	@10	@100	@10	@100	@10	@100
RWRDoc	0.5877	0.7215	0.4189	**<u>0.5296</u>**	0.4119	**<u>0.5845</u>**	0.3346	**<u>0.5163</u>**	0.4348	**<u>0.5643</u>**
Residual	-6.86%	-0.42%	-2.05%	0%	-1.33%	+7.03%	-3.43%	+5.49%	-2.57%	+1.38%

- Not much good ranking capability
  - esp. top-10 ranking quality is always inferior to the best state-of-the-art
- For harder task, top-100 ranking quality is fairly good.
  - <u>RWRDoc can pus-up relevant entities in lower position</u>

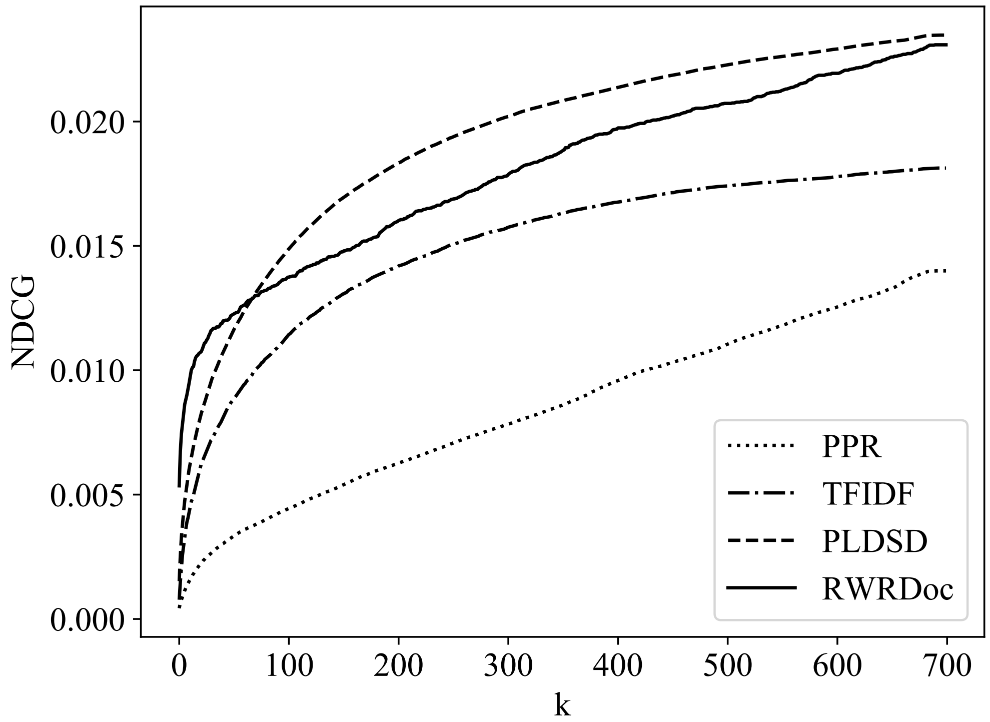
Recommendation Task

- LD is used as auxiliary info. to improve recommender system performance [2, 13]
  - Taking semantic similarity of items into account
    - [13] measures it by personalized PageRank.
    - [2] is based on commonality of neighbours in LD.
    - A baseline is cosine similarity b/w TFIDF vectors.
- Benchmark: HetRec 2011 dataset*1
  - Listening list of artists in Last.FM
  - To connect with LD, mapping data*2 is also used.
- Quality measure: NDCG

*1 https://grouplens.org/datasets/hetrec-2011/

*2 http://sisinflab.poliba.it/semanticweb/lod/recsys/datasets/

Accuracy of Recommendation



- RWRDoc is better in earlier rankings but PLDSD is better in later rankings.

Findings from Rec. Task

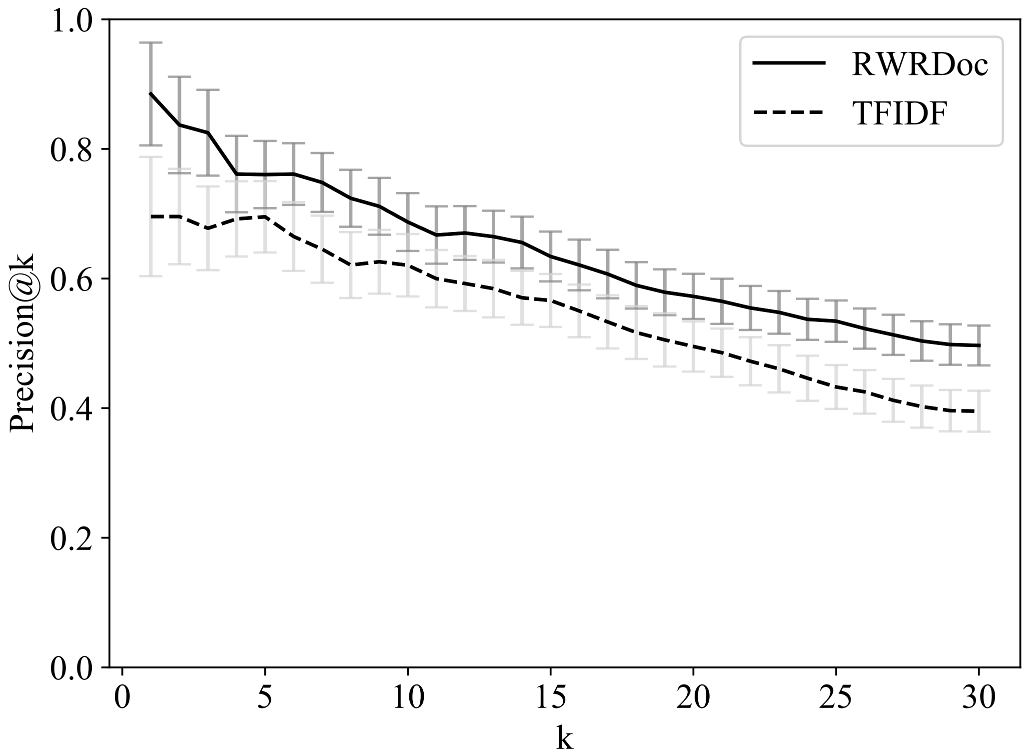
- RWRDoc is an in-between method of text-only method (i.e., TFIDF) and topology-only method (i.e., PPR and PLDSD).
- RWRDoc is superior to the both methods.
  - Taking both text and topology into account can improve recommendation quality.
- Improving later ranking is an issue.
  - More sophisticated topology-based approach (like PLDSD) should be considered.

Summarization Task

- For each entity, show top-30 representative terms in the representation and human judges evaluate whether the term is relevant.
  - Baseline: TFIDF (minimal representation)
  - RWRDoc representation
- Quality measure: precision@k

Precision of Summary Terms

- Figure
  - Line: average
  - Error bar: deviation
- RWRDoc is superior to the baseline



Examples of Representations

(a) Hideyoshi Toyotomi

RWRDoc	Rel.	TFIDF	Rel.
joseon	✓	period	
dynasty	✓	samurai	✓
period		unifier	✓
samurai	✓	momoyama	✓
unifier	✓	ieyasu	✓
momoyama	✓	nobunaga	✓
ieyasu	✓	daimyo	✓
nobunaga	✓	liege	✓
daimyo	✓	sengoku	✓
liege	✓	legacies	

(b) Nagoya

RWRDoc	Rel.	TFIDF	Rel.
japan	✓	chky	
chky		japan	✓
chunichi	✓	metropolitan	✓
wii		largest	
metropolitan	✓	area	
chunichidragonzu	✓	kitakyushu	
doala	✓	chubu	✓
chunichi	✓	city	✓
region		honshu	✓
city	✓	aichi	✓

- Rel.: relevance judgement
- Shaded: only appear in top-30 of the rep.

Remarks: pros and cons

- Pros
  - RWRDoc successfully incorporates related facts into entity representations.
  - RWRDoc achieves (not always significant but) better results in various tasks.
- Cons
  - RWRDoc fails to incorporate relationship information (i.e., predicates) into entity representation.

Conclusion

- RWRDoc
  - Combination of minimal representations of entities and RWR
    - RWR measure reachability to relevant entities.
    - Weighted sum of minimal representations in terms of RWR scores provides representations.
  - Experimental evaluation reveals pros and cons of RWRDoc
- Future direction
  - Taking predicate information into account to improve the representations