CS:4980:0004
MINING AND LEARNING ON LARGE NETWORKS

Joyce Yang

Lecture 17: E-Commerce

Fall 2020

Outline

- B. Adhikari et al. Mining E-Commerce Query Relations using Customer Interaction Networks. (Links to an external site.) WWW 2018.
- D. Xu et al. Product Knowledge Graph Embedding for E-Commerce (Links to an external site.). WSDM 2020

Mining E-Commerce Query Relations using Customer Interaction Networks

[Mining E-Commerce Query Relations using Customer Interaction Networks]

User Engagement

- Useful information behind the search engine logs.

- Begin with query submission
- List of results are produced
- Based on the result
  - Click - satisfied
  - Reformulate the query – not satisfied

Mining E-Commerce Query Relations using Customer Interaction Networks

Engagement Data

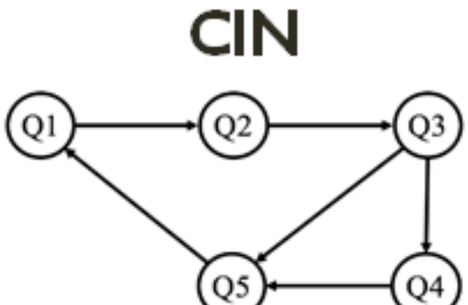
Engagement log

Query	Item	Time
Q1	I1	07 :00
Q2	I2	07:15
Q3	I3	07:20

Query	Item	Time
Q2	I3	09 :00
Q3	I4	09:15
Q4	I3	09:30

Query	Item	Time
Q2	I3	17 :00
Q3	I4	17:15
Q5	I5	17:30

Query	Item	Time
Q4	I3	17 :00
Q5	I4	17:15
Q1	I2	17:30



APPLICATION 1: INTENT BASED QUERY CLUSTERING

APPLICATION 2: ITEM RECOMMENDATION

APPLICATION 3: CRITICAL QUERIES

Mining E-Commerce Query Relations using Customer Interaction Networks

Customer Interaction Networks (CIN)

- Query Reformulation Network (QRN)
  - Query-to-Query relations
  - Nodes:Queries string
  - Egde: Reformulation
- Item Clicked Network (ICN)
  - Query-to-Item relations
  - Nodes:Queries/Items
  - Edges:Click
- Composite Click Network (CCN)
  - Item Click + Query Reformulation = Compo ork
- Cover Network(CN)

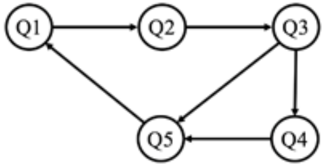
Engagement log

Query	Item	Time
Q1	I1	07 :00
Q2	I2	07:15
Q3	I3	07:20

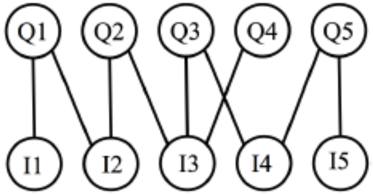
Query	Item	Time
Q2	I3	09 :00
Q3	I4	09:15
Q4	I3	09:30

Query	Item	Time
Q2	I3	17 :00
Q3	I4	17:15
Q5	I5	17:30

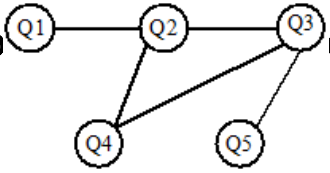
Query	Item	Time
Q4	I3	17 :00
Q5	I4	17:15
Q1	I2	17:30



Query Reformulation Network



(c) *Item Click* Network



Cover Network

[Mining E-Commerce Query Relations using Customer Interaction Networks]

Properties of CIN

Network	# Nodes	# Edges	Degree Dist.	Assortativity	Diameter	Average CC	GCC
QRN	2.11 M	2.14 M	PL/LN	None	94	0.05	No
ICN	5.4 M	18.4 M	LN	None	37	0.12	No
CCN	6.3 M	20.5 M	PL	None	36	0.17	No
CN	785 K	71 M	LN	Positive	13	0.76	Yes

[Mining E-Commerce Query Relations using Customer Interaction Networks]

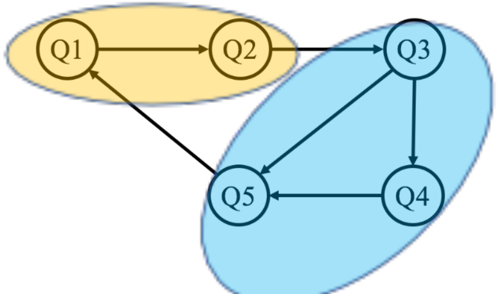
Application 1: Intent Based Query Clustering

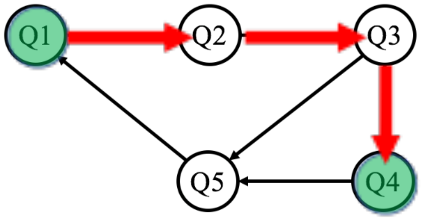
Informal Problem 1. *Query Clustering*
Given: *A Query Reformulation network $G(Q, E, W)$, and an integer $k \in Z$.*
Find: *A k partition of Q, such that each partition contains queries with the same intent.*

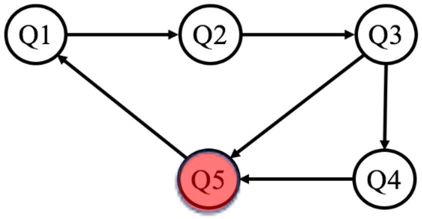
i) How is intent defined in term of graph structure?
i) High in-degree nodes → general queries
ii) How to measure 'closeness' between two queries in terms of intent?
i) Shortest paths → closer relationship

"Phone" "Mobile" "TV" "Television"

"TV" "Television"
"Phone" "Mobile"







Mining E-Commerce Query Relations using Customer Interaction Networks

Define Problem

Problem 1. *Given a query reformulation network* $G(Q, E)$ *and an integer* k, *identify a set* $S^* = \{S_1, S_2, ..., S_k\}$ *of the general query nodes and set of partitions* $C^* = \{C_1, C_2, ..., C_k\}$, *such that* $S_i \in C_i$ *and*

$$S^*, C^* = \arg\min_{S,C} J(S, C) = \arg\min_{S,C} \left[\boxed{\left(\sum_{v \in V} d(v, s(v)) \right)} \boxed{\left(\sum_{s \in S} \frac{1}{\theta^i(s)} \right)} \right]$$

Short Distance to center

High in-degree center

Mining E-Commerce Query Relations using Customer Interaction Networks

Our Method: HubQExpansion

- Idea: Leverage newly discovered properties of QRN.

1. Detect Communities in each component(No GCC)
2. Assign 'hubs' to each Community (Heavy Tailed Distributions)
3. Use BFS to expand communities (Low Density)

Algorithm 1 HubQExpansion

Require: *Query Reformulation* network $G(Q, E, W)$, number of communities k

Ensure: k disjoint partitions of Q

```
Partition P = ∅
for each connected component G_i(Q_i, E_i, W_i) in G do
    k_i = |Q_i| / |Q|
    Temp set S = ∅
    for node v in k_i nodes in Q_i with highest in-degree do
        S = S ∪ {v}
        Assign nodes in Q_i to nodes in S using BFS
    P = P ∪ S
return P
```

Baseline Methods

- **BigClam**[Yang+, 2013]: Leverage latent relevance to find overlapping clusters
- **Louvian**[Blondel+, 2008]: Modularity based community detection method
- **LouvSmall**: Modify Louvian to generate smaller communities
- **Star**: Generate Star Shaped Communities

[Mining E-Commerce Query Relations using Customer Interaction Networks]

Evaluation

- AIH: Average Intent Homogeneity
  - Intuitively measures category-based precision

$$CIH(C) = 2 * \frac{\sum_{q_i, q_j \in C} \delta(PC(q_i), PC(q_j))}{|C| \times |C-1|} \Rightarrow AIH_{cat} = \frac{\sum_{C \in P} CIH(C)}{|P|}$$

- AIS: Average Inverse Spread
  - Intuitively measures category-based recall

- F1: Harmonic mean of AIH and AIS

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}} = \left(\frac{\sum_{i=1}^{n} x_i^{-1}}{n} \right)^{-1}.$$

Mining E-Commerce Query Relations using Customer Interaction Networks

Performance

More results in the paper

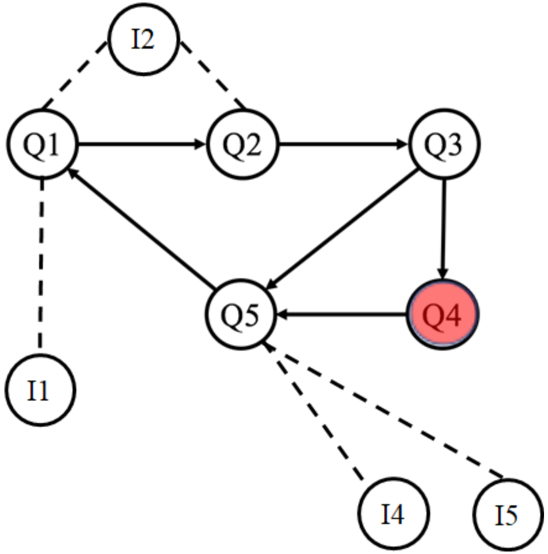
Method	AIH_{cat}	AIS_{cat}	$F1_{cat}$	$J(\times 10^6)$
LOUVIAN	0.26	0.11	0.15	19.7
COMLOUVIAN	0.07	0.33	0.12	118.7
LOUVIANSMALL	0.39	0.08	0.13	0.73
STAR	0.38	0.12	0.18	3.01
BIGCLAM	0.14	0.21	0.17	17.7
HUBQEXPANSION	0.37	0.14	**0.20**	**0.54**

HubQExpansion outperforms the baselines

[Mining E-Commerce Query Relations using Customer Interaction Networks]

Application 2: Item Recommendation

- How to recommend items for queries with no/little engagement data?



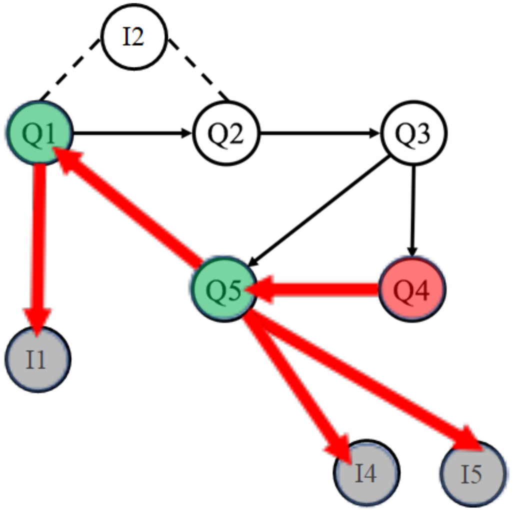
Composite Click Network

How to recommend items for query 4?

[Mining E-Commerce Query Relations using Customer Interaction Networks]

Idea

- Composite Click Network
- Leverage Random Walk
  - Treat a query (eg Q4) as a source node
  - Items as sink nodes
  - Start multiple random walks from Q4
  - Recommend the products where most of the random walks end
- Edge weight (unnormalized)
  - Query q_1 to q_2 edge: $w(q_1, q_2) = \frac{c(q_1, q_2)}{c(q_1)}$
  - Query q to product i edge: $w(q, i) = \frac{c(q, i)}{c(q)}$,



Experiments and Results

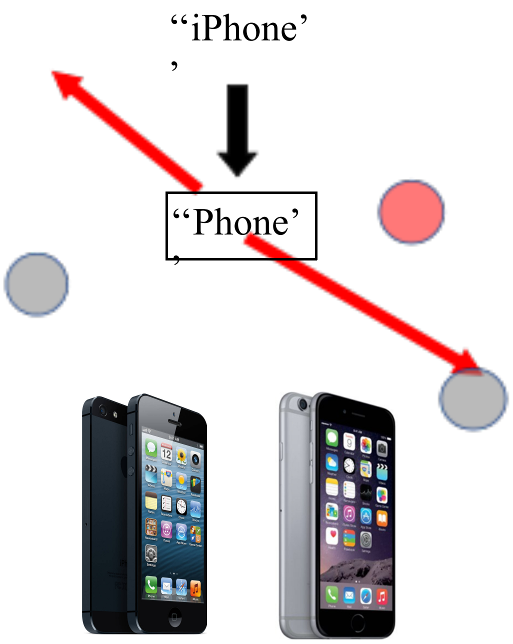
- Experiment:
  - A/B test
- Result:
  - 34% improvement in NDGC over current Walmart.com search engine!

[Mining E-Commerce Query Relations using Customer Interaction Networks]

Application 3: Critical Queries

Which queries have the highest cumulative impact on others?

Leverage Query Reformulation Network.

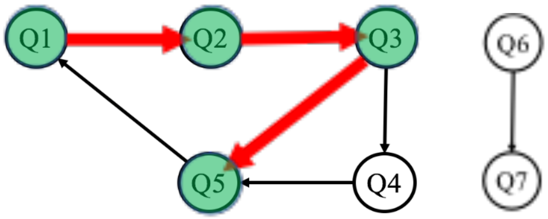


RUN: Randomized User Navigation model

- Starts from arbitrary Node

(1) With probability p_t, the process terminates

(2) With probability $1 - p_t$, we jump from current node v to a query node u, such that $(v, u) \in E$, with probability
$$p_j = \frac{w(v,u)}{\sum_{(v,a)\in E} w(v,a)}$$



Reformulation Log:
Q1, Q2, Q3, Q5

Mining E-Commerce Query Relations using Customer Interaction Networks

Define Problem

-
- $\phi(T)$: the expected number of ti

refo ... ing to

Problem 2. *Critical Queries*

Given: *A Query Reformulation network $G(Q, E, W)$, and budget $k \in Z$.*

Find: *a set of nodes $\mathcal{T}^* = \{q | q \in Q\}$, such that $|\mathcal{T}^*| = k$ and*

$$\mathcal{T}^* = \arg\max_{\mathcal{T}} \phi(\mathcal{T})$$

NP-Hard

Lemma 7.1. *$\phi(\cdot)$ is sub-modular and monotonous.*

$$\phi(\mathcal{A} \cup \{v\}) - \phi(\mathcal{A}) \geq \phi(\mathcal{B} \cup \{v\}) - \phi(\mathcal{B})$$

[Mining E-Commerce Query Relations using Customer Interaction Networks]

Our Method: CriticalQueries

Algorithm 2 CRITICAL-QUERIES (GR)

Require: *Query Reformulation* network $G(Q, E, W)$, termination probability p_t, number of iterations of RUN l, and budget k

Ensure: Best set of nodes $\mathcal{T}$

- $\mathcal{T} = \emptyset$
- **for** $i = 1$ to k **do**
    - $g_v = 0$ for all $v \in V \setminus \mathcal{T}$
    - **for** $i = 1$ to R **do**
        - Sample G' based on p_t and l
        - compute $\phi_{G'}(\mathcal{T})$
        - **for** $v \in V \setminus \mathcal{T}$ **do**
            - $g_v += \phi_{G'}(\{v\})$
    - $g_v = g_v / R$ for all $v \in V \setminus \mathcal{T}$
    - $\mathcal{T} = \mathcal{T} \cup \{\arg\max_v (g_v)\}$
- return $\mathcal{T}$

Baseline Methods

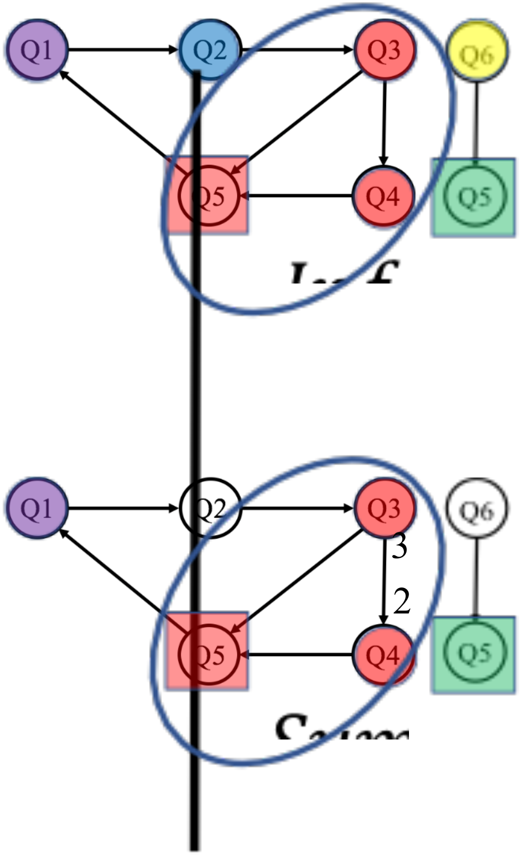
One can leverage many methods/definitions

- MostFreq (MF) : Queries most frequent
- SessionFreq(MS): Queries Appearing in Most Sessions
- PageRank(PR): Queries with highest PageRank
- EigCentrality(EigC): Queries with highest Eigenvector Centrality

Evaluation

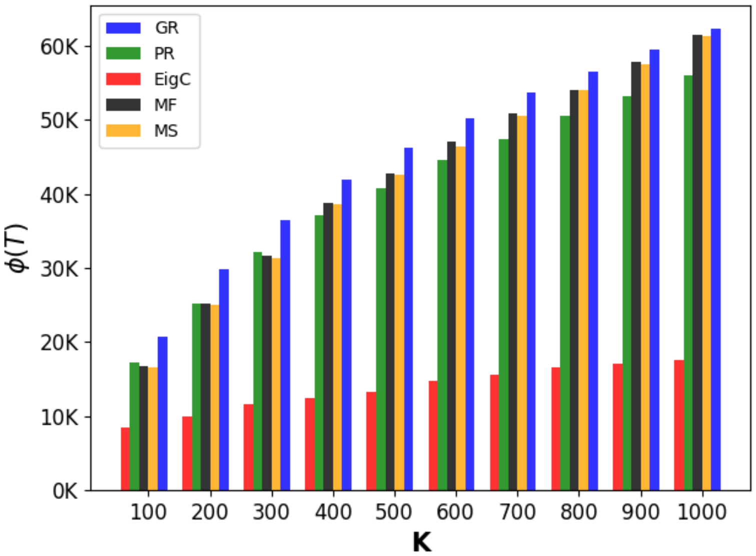
- **INFluenced Queries (InfQ):**
  - **Number** of **related queries** within some radius

- **Sum of related Items (SumI)**
  - **Number** of **related items** shared with other queries within some radius

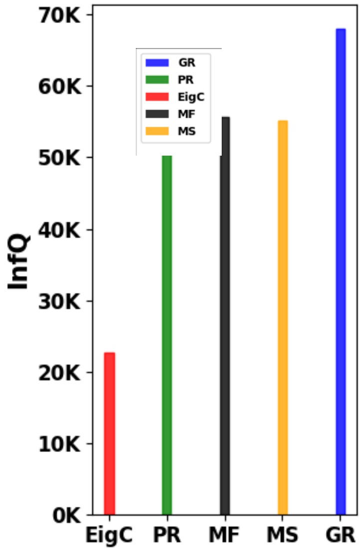


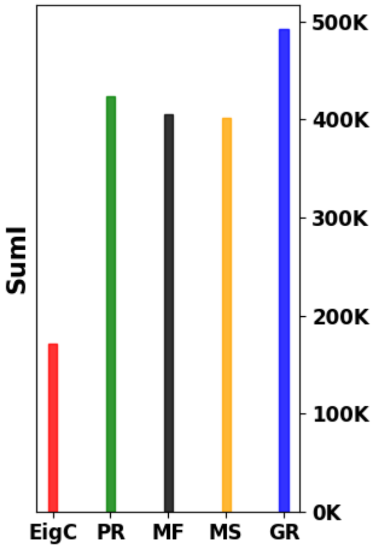
Results

Higher is better



Our metho



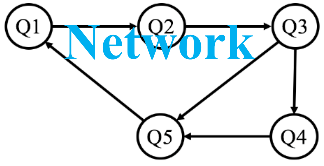


Our method has the best usability

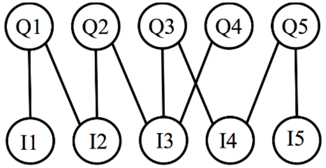
Conclusion

- **CINs are useful**
  - They have useful properties
  - A1: Intent Based Query Clustering
  - A2: Item Recommendation
  - A3: Critical Queries
- **Formulation and Methods**
  - exploit CINs properties
  - scalable
- **Future Work**
  - Add content
  - Add query attributes
  - Other applications like type-ahead, query

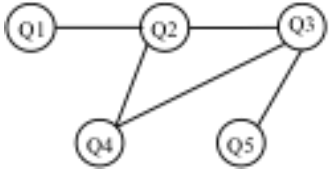
Query Reformulation Network



Item Click Network



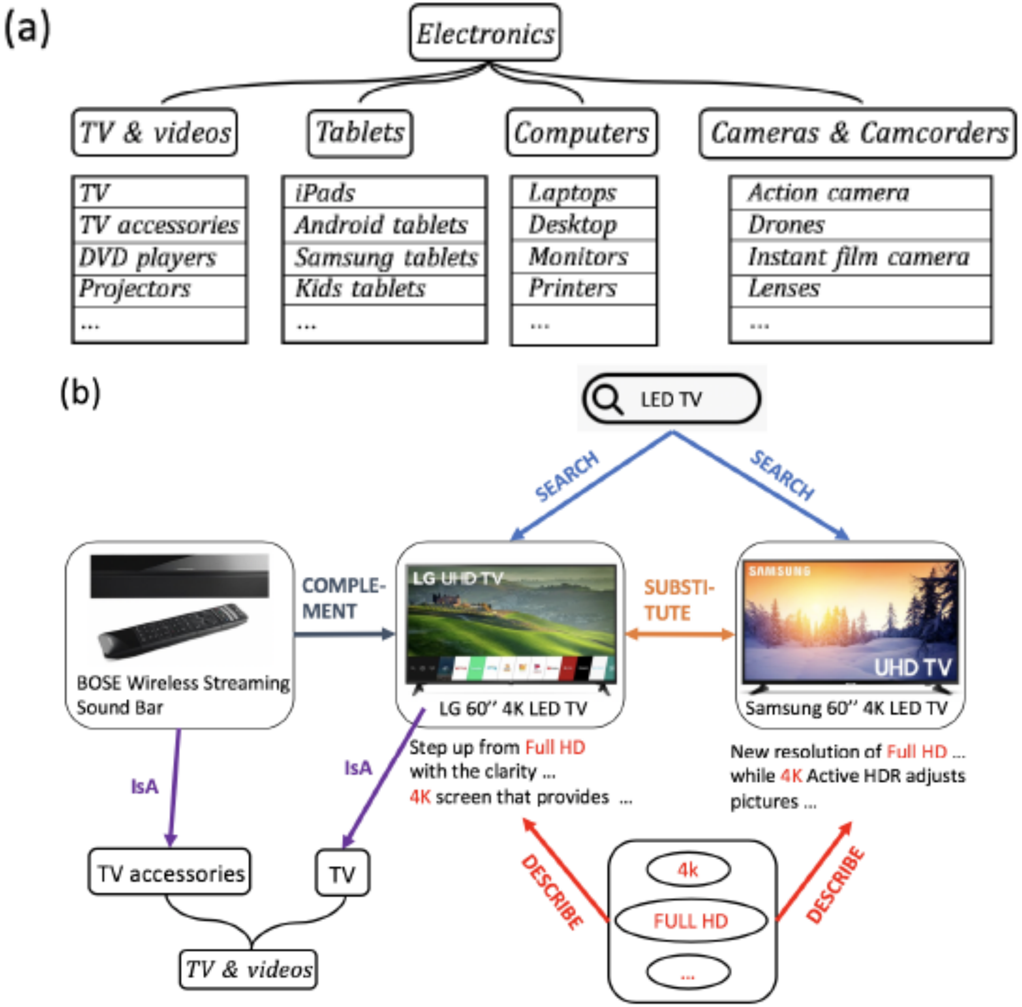
Cover Network



Product Knowledge Graph Embedding for E-Commerce

Background

- Product relations
  - Complement(co-buy), co-view and substitute …
- Useful information:
  - Product description
  - Search activities
- Extension:
  - Complement : AddOn, AccessoryTo, part of.
  - Description: HasAttribute, Brand, Name



Knowledge Graph **VS** Product Knowledge Graph

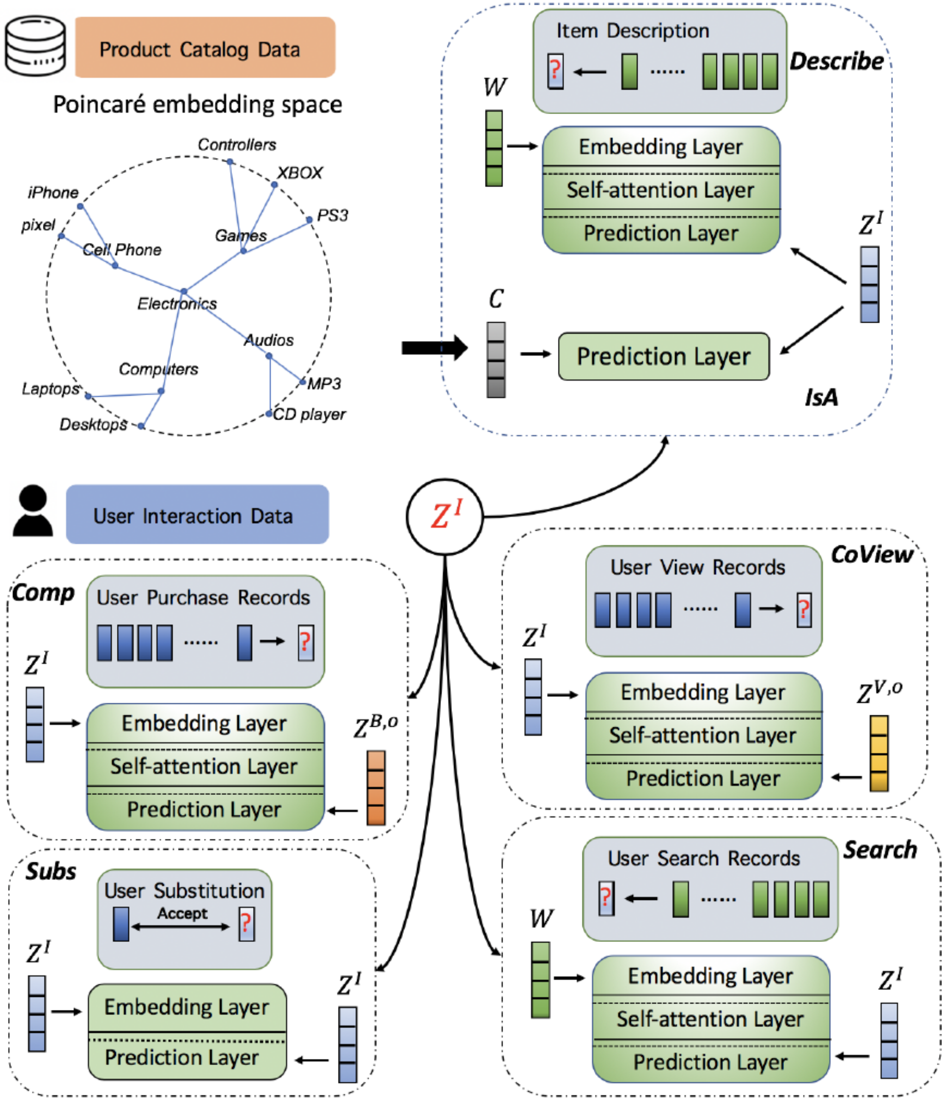
	Knowledge Graph	**Product Knowledge Graph**
Data Source	Triplet form (head entity, relation, tail entity)	Multiple ways Product catalog info, raw user-product interaction records
Model Assumption	Well-established & plausible	X
Quantity of relation types	thousands	Complement, co-view, substitute, describe, search and IsA
Semantics of relation	Simple (1-to-N and 1-to-1)	Complex (N-toN)
Additional information	Yes	Yes
Logic rules	First order Horn clause	Propagation rules
Downstream tasks	KG completion tasks: Relation extraction and question-answering	Relation extraction and question-answering + Recommender system

Product Knowledge Graph Embedding for E-Commerce

Method: PKG embedding

Notation	Description
$d \in \mathbb{N}$	entity embedding dimension
$l \in \mathbb{N}$	the maximum sequence length for predicting target entity
$\mathcal{I}, \mathcal{W}, \mathcal{C}$	item (product), word and category label set
$\mathcal{B}, \mathcal{V}, \mathcal{S}$	customer session-based co-buy, co-view and substitution acceptance records, for instance, $\mathcal{B}_i = (I_{i_1}, I_{i_2}, \ldots, I_{\lvert\mathcal{B}_i\rvert})$, $\mathcal{S}_j = (I_{j_1}, I_{j_2})$
$\mathcal{Q}, \mathcal{D}, \mathcal{L}$	query (search), description, category labels for products, i.e. $\mathcal{Q}_{I_j} \subset \mathcal{W}$, $\mathcal{D}_{I_j} \subset \mathcal{W}$, $\mathcal{L}_{I_j} \subset \mathcal{C}$
$\mathrm{Z}^I \in \mathbb{R}^{\lvert\mathcal{I}\rvert \times d}$	product "input" embedding similar to that of *word2vec*, e.g. $\mathrm{Z}^I_{I_j}$ denotes product "input" embedding for item I_j
$\mathrm{Z}^{B,O}, \mathrm{Z}^{V,O} \in \mathbb{R}^{\lvert\mathcal{I}\rvert \times d}$	product "output" embeddings for modelling co-buy and co-view records
$\mathrm{W} \in \mathbb{R}^{\lvert\mathcal{W}\rvert \times d}$	word entity embeddings
$\mathrm{C} \in \mathbb{R}^{\lvert\mathcal{C}\rvert \times d}$	category entity embeddings
$\mathrm{P} \in \mathbb{R}^{l \times d}$	positional encoding matrix for self-attention

Table 1: Notation



Pre-Method: Relation translation

- The distributed representation of words are learnt under the hypothesis that words which appear in a similar context have a similar representation
- word2vec: King is to men as queen is to women

$$z_{\text{king}} - z_{\text{men}} \approx z_{\text{queen}} - z_{\text{women}}, \quad \Rightarrow \quad \begin{cases} z_{\text{king}} \approx z_{\text{men}} + z_{\text{royal}} \\ z_{\text{queen}} \approx z_{\text{women}} + z_{\text{royal}}. \end{cases}$$

THEOREM. *If "entity y_1 is to entity x_1 as entity y_2 is to x_2", then:*

$$z_{y_1} = z_{x_1} + (z_{y_2} - z_{x_2}) + \epsilon$$

$$z_{\text{remote control}} + z_{\text{AccessoryTo}} \approx z_{\text{TV}} \qquad z_{\text{AccessoryTo}} \equiv z_{\text{Xbox}} - z_{\text{handle}}$$

Pre-Method: word2vec

- Score function:

$$S = \sum_i \sum_{j \in \text{Context}(i,c)} \log p(e_i | e_j),$$

- Probability term:

$$p(e_i|e_j) = \frac{\exp\left((\mathbf{z}_i^O)^\top(\mathbf{z}_i^I)\right)}{\sum_k \exp\left((\mathbf{z}_k^O)^\top(\mathbf{z}_i^I)\right)},$$

$\mathbf{Z}^I \in \mathbb{R}^{\lvert\mathcal{I}\rvert \times d}$	product "input" embedding similar to that of *word2vec*, e.g. $\mathbf{Z}^I_{\mathcal{I}_j}$ denotes product "input" embedding for item $\mathcal{I}_j$	$\mathbf{Z}^{B,O}, \mathbf{Z}^{V,O} \in \mathbb{R}^{\lvert\mathcal{I}\rvert \times d}$	product "output" embeddings for modelling co-buy and co-view records

Method: Modelling ***substitute*** Relation

- Propagation rule:
  - *substitutable products are more likely to have similar relations with other entities.*

$$S_{\text{sub}} = \sum_{(e_1, e_2) \in \mathcal{S}} \log p(e_1, e_2) \equiv \sum_{(e_1, e_2) \in \mathcal{S}} \log \frac{\exp\left((\mathbf{Z}^{I}_{e_1})^\top \mathbf{Z}^{I}_{e_2}\right)}{\sum_{i \in \mathcal{I}} \exp\left((\mathbf{Z}^{I}_{i})^\top \mathbf{Z}^{I}_{e_2}\right)}. \tag{3}$$

$\mathbf{Z}^I \in \mathbb{R}^{\lvert\mathcal{I}\rvert \times d}$	product "input" embedding similar to that of *word2vec*, e.g. $\mathbf{Z}^I_{I_j}$ denotes product "input" embedding for item I_j

Pre- Method: Self-attention Mechanism

- To break down the **noise** issue

[. . . , soap, detergent, toothbrush, towel → toothpaste, . . .].

{The strawberry ice cream featured by Haagan-Dazs is the
marriage of sweet summer strawberries to cream and . . .},

Method: Self-attention Mechanism

- Embedding layer for self-attention

$$\mathrm{E}_{\mathrm{e}}^{I} = [\mathrm{Z}_{e_1}^{I} + \mathrm{P}_1, \ldots, \mathrm{Z}_{e_1}^{I} + \mathrm{P}_l]^{\top} \in \mathbb{R}^{l \times d} \quad \Rightarrow \quad \mathrm{E}_{\mathrm{e}}^{O} = [\mathrm{Z}_{e_1}^{O} + \mathrm{P}_1, \ldots, \mathrm{Z}_{e_1}^{O} + \mathrm{P}_l]^{\top} \in \mathbb{R}^{l \times d}.$$

- Self-attention layer

$$\mathrm{Attn}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \mathrm{softmax}\left(\frac{\mathrm{QK}^{\top}}{\sqrt{d}}\right)\mathrm{V}, \quad \Rightarrow \quad \mathrm{H} = \mathrm{Attn}(\mathrm{E}^{I}, \mathrm{E}^{O}, \mathrm{E}^{I}).$$

$$\mathrm{H}_i = \sum_j \alpha_{ij} \times \mathrm{E}_j^{I}, \quad \alpha_{ij} \equiv (\mathrm{E}_i^{I})^{\top} \mathrm{E}_i^{O}$$

$$\mathrm{FFN}(\mathrm{E}_i) \equiv \mathrm{ReLU}(\mathrm{E}_i \Theta_1 + \mathrm{b}_1)\Theta_2 + \mathrm{b}_2,$$
$$\mathrm{F}_i^{I} = \mathrm{FFN}(\mathrm{E}_i^{I}), \ \mathrm{F}_i^{O} = \mathrm{FFN}(\mathrm{E}_i^{O}), \ \forall i \in \{1, \ldots, l\}. \quad \Rightarrow \quad \mathrm{H} = \mathrm{Attn}(\mathrm{e}) \equiv \mathrm{Attn}(\mathrm{F}_{\mathrm{e}}^{I}, \mathrm{F}_{\mathrm{e}}^{O}, \mathrm{E}_{\mathrm{e}}^{I}), \mathrm{H} \in \mathbb{R}^{l \times d}.$$

Method: Self-attention Mechanism

Word2vec probability

$$p(e_i|e_j) = \frac{\exp\left((\mathbf{z}_i^O)^\top(\mathbf{z}_i^I)\right)}{\sum_k \exp\left((\mathbf{z}_k^O)^\top(\mathbf{z}_i^I)\right)},$$

- Prediction layer

purchase and view data

$$\log p(e_{l+1}|\mathbf{e}) = (\mathbf{Z}^O_{e_{l+1}})^\top \mathrm{Attn}(\mathbf{e}) - \log \sum_{i\in\mathcal{I}} \exp\left((\mathbf{Z}_i^O)^\top \mathrm{Attn}(\mathbf{e})\right)$$

$$= (\mathbf{Z}^O_{e_{l+1}})^\top \sum_{j=1}^{l} \alpha_{ij}\mathbf{Z}^I_{e_j} - \log \sum_{i\in\mathcal{I}} \exp\left((\mathbf{Z}_i^O)^\top \sum_{j=1}^{l} \alpha_{ij}\mathbf{Z}^I_{e_j}\right) + C,$$

description and search data

$$\log p(e_{l+1}|\mathbf{e}) = (\mathbf{Z}^I_{e_{l+1}})^\top \sum_{j=1}^{l} \alpha_{ij}\mathbf{W}^I_{e_j} - \log \sum_{i\in\mathcal{I}} \exp\left((\mathbf{Z}_i^I)^\top \sum_{j=1}^{l} \alpha_{ij}\mathbf{W}^I_{e_j}\right) + C,$$

- Score function

$$S_{\text{complement}} = \sum_{(e_{l+1},\mathbf{e})\in\mathcal{B}} \log p(e_{l+1}|\mathbf{e}).$$

$\mathbf{W} \in \mathbb{R}^{\lvert\mathcal{W}\rvert\times d}$	word entity embeddings

Method: Poincaré Embedding for Category Hierarchy and IsA Relation

$$p(c_1|c_2) = \frac{\exp\left(-d_{\text{Poincaré}}(c_1, c_2)\right)}{\sum_{c \in C} \exp\left(-d_{\text{Poincaré}}(c, c_2)\right)}.$$

$$S_{\text{IsA}} = \sum_{i \in |\mathcal{I}|} \sum_{j \in \mathcal{L}_i} \log \frac{\exp\left(\mathbf{C}_j^\top \mathbf{Z}_i^I\right)}{\sum_{c \in C} \exp\left(\mathbf{C}_c^\top \mathbf{Z}_i^I\right)}.$$

Method: Multitask Training

- How to combine all these score functions into an overall score for optimal results?
- Training epoch:
  - Randomly select a task
  - And a batch of dataset
  - [weighted with proportional]
  - Repeat until no improvement

Task	$p(t \mid h)$ / $p(t \mid \mathbf{h})$
substitute	$\propto \exp\left((\mathrm{Z}_t^I)^\intercal \mathrm{Z}_h^I\right), h, t \in \mathcal{I}$
complement	$\propto \exp\left((\mathrm{Z}_t^{B,O})^\intercal \mathrm{Z}_h^I\right), h, t \in \mathcal{I}$
co-view	$\propto \exp\left((\mathrm{Z}_t^{V,O})^\intercal \mathrm{Z}_h^I\right), h, t \in \mathcal{I}$
search describe	$\propto \exp\left((\mathrm{Z}_t^I)^\intercal \mathrm{Attn}(\mathbf{h})\right), \mathbf{h} \subset \mathcal{W}, t \in \mathcal{I}$
Isa	$\propto \exp\left((\mathrm{Z}_t^I)^\intercal \mathrm{C}_h^I\right), h \in \mathcal{C}, t \in \mathcal{I}$
recommend	$\propto \exp\left((\mathrm{Z}_t^{B,O} + \mathrm{Z}_t^{V,O})^\intercal \mathrm{Attn}(\mathbf{h})\right), \mathbf{h} \subset \mathcal{I}, t \in \mathcal{I}$

Table 2: Prediction for each task (relation) according to trained embeddings.

Experiments & Results

Q1: Is the proposed multi-task learning schema reasonable?

Q2: Other than knowledge completion, how does the PKG embedding benefit downstream e-commerce tasks?

Q3: Why KG embedding methods fail to work when directly applied to raw e-commerce dataset?

Q4: If a product relation graph is available for KG embedding methods, can the proposed approach still outperform the baselines?

Dataset

- Session data
- Substitution data
- Preprocess
- Product relation graph(PRG)

products	The dataset contains ~140,000 common grocery products covering a broad range from food to appliances
description	Each product is provided with a short description (containing name and brand) as shown on the website. Usually the descriptions have 20 - 100 words.
category hierarchy	Each product is assigned to a cateogry hierarchy in the form of {subcategory, category, department, super-department}, and there are 1,198 subcategories, 228 categories, 28 departments and 9 super-departments.

Table 3: Summary of product catalog data.

Baseline Methods

KG completion tasks:

- TransE (No PRG & With PRG)

$$d_r(\mathbf{h}, \mathbf{t}) = \sum -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$$

- Trans

$$d_r(\mathbf{h}, \mathbf{t}) = \sum -\|\mathbf{h}_\perp^r + \mathbf{r} - \mathbf{t}_\perp^r\|$$

- TransR (With PRG)
- TransD (No PRG & With PRG)

$$d_r(\mathbf{h}, \mathbf{t}) = \sum \mathbf{h}^\top \mathbf{M}_r \mathbf{t},$$

- RESCAL (With PRG)
- D ... ith PRG)

$$d_r(\mathbf{h}, \mathbf{t}) = \sum \mathrm{Re}\left(\mathbf{h}^\top \mathrm{diag}(\mathbf{r}) \bar{\mathbf{t}}\right),$$

- ComplEx (With PRG)

Recommendation:

- Factorization Machine (FM)
- Bayesian Personalized Ranking (BPR)
- Prod2vec
- Triple2vec

Evaluation

- Knowledge completion:
  - Link prediction – usefulness
    - **HIT@10**: top-10 hitting rate
    - **NDCG@10:** normalized discounted cumulative ga
  - Entity classi
    - Micro-F1
    - Macro-F1
- Search ranking:
  - R@10: top-10 recall
  - MAP@10: mean average precision
- Recommendation:
  - HIT@10, NDCG@10

$$nDCG_p = \frac{DCG_p}{IDCG_p} \quad DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

Where

$$IDCG_p = \sum_{i=1}^{|REL_p|} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

$$\text{Micro F1-score} = 2 * \frac{\text{Micro-precision} * \text{Micro-recall}}{\text{Micro-precision} + \text{Micro-recall}}$$

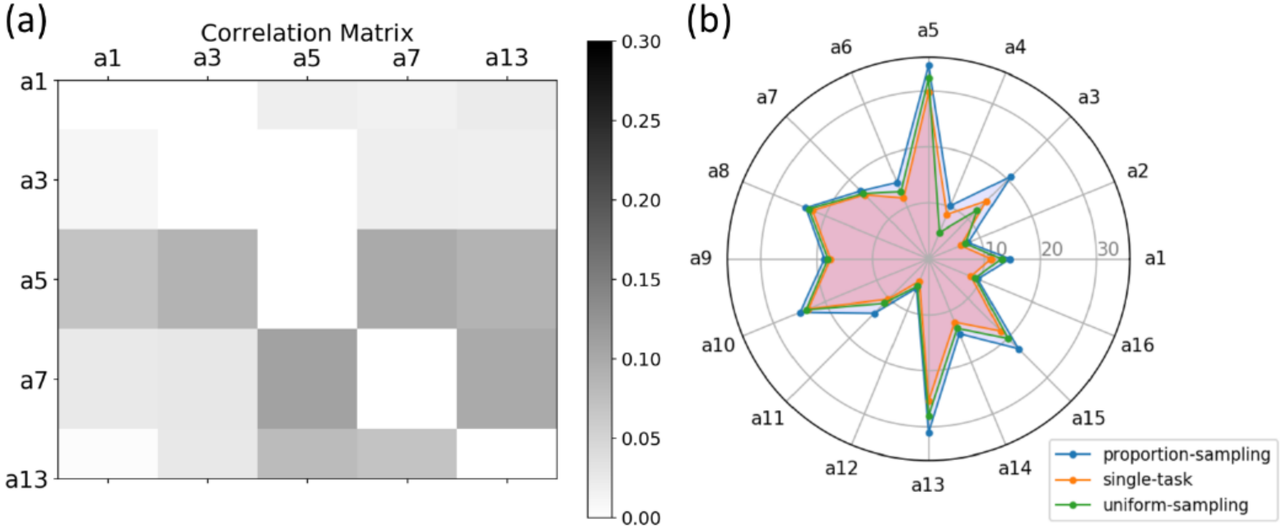
$$\text{Macro F1-score} = \frac{1}{N} \sum_{i=0}^{N} \text{F1-score}_i$$

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

Results

- **Q1:**Is the proposed multi-task learning schema reasonable?
- our approach is benefiting from the propagation rule



Results

- **Q2 :** Other than knowledge completion, how does the PKG embedding benefit downstream e-commerce tasks?
- the PKG embeddings learned by our approach can also benefit downstream tasks such as search ranking and recommendation

Model	FM	BPR	prod2vec	triple2vec	Our approach
Hit@10 (a11)	4.24	7.65	6.39	*11.30*	***13.72***
NDCG@10 (a12)	1.85	3.17	2.26	*4.83*	***5.79***

Task	**Encountered queries**		**New queries**	
Metric	**R@10** (a13)	**MAP@10** (a14)	**R@10** (a15)	**MAP@10** (a16)
TransE	8.74	3.26	5.11	2.62
TransH	10.43	4.28	6.85	2.79
TransR	15.82	6.77	10.33	4.09
TransD	13.69	6.17	9.50	4.24
RESCAL	7.71	2.98	4.25	1.93
DistMult	19.72	8.43	11.71	5.02
ComplEx	*21.58*	*10.04*	*12.46*	*5.15*
Our approach	***30.99***	***14.46***	***22.71***	***9.53***

Results

- **Q3:** Why KG embedding methods fail to work when directly applied to raw e-commerce dataset?
- The fact that KG embeddings have subpar results when not using PRG suggests they rely heavily on well-established facts which are absent in the raw dataset

Task	Link prediction						Product classification			
Relation	complement		co-view		substitute		IsA (*category*)		IsA (*department*)	
Metric	**Hit@10** (a1)	**NDCG@10** (a2)	**Hit@10** (a3)	**NDCG@10** (a4)	**Hit@10** (a5)	**NDCG@10** (a6)	**micro-F1** (a7)	**macro-F1** (a8)	**micro-F1** (a9)	**macro-F1** (a10)
TransE (No PRG)	1.84	1.06	3.27	2.11	12.04	6.56	42.33	69.44	51.72	76.53
TransD (No PRG)	1.97	1.08	3.51	2.24	13.69	6.90	41.82	67.65	50.29	75.45
DistMult (No PRG)	3.47	1.88	6.58	3.41	20.64	9.96	53.75	74.69	61.42	80.73
TransE (With PRG)	3.65	1.82	6.95	3.90	30.22	13.41	45.43	74.93	55.89	81.81
TransH (With PRG)	4.13	1.79	6.88	2.89	30.37	13.56	41.94	64.09	50.12	72.95
TransR (With PRG)	6.06	2.35	8.17	3.43	31.25	14.88	46.37	72.74	53.95	74.11
TransD (With PRG)	4.26	1.95	7.03	2.97	20.71	9.86	50.36	71.02	59.62	82.43
RESCAL (With PRG)	1.64	0.97	1.63	0.87	12.46	5.76	62.89	86.27	72.27	90.97
DistMult (With PRG)	5.69	2.47	9.64	4.05	30.64	12.25	*68.25*	*94.23*	72.09	92.94
ComplEx (With PRG)	*7.81*	*3.36*	*12.38*	*5.77*	*31.25*	*12.60*	67.46	94.02	*72.54*	*97.71*
Our approach	***14.53***	***7.67***	***20.84***	***10.26***	***34.58***	***14.77***	***68.62***	***95.17***	***74.61***	***99.60***

Results

- **Q4:** If a product relation graph is available for KG embedding methods, can the proposed approach still outperform the baselines?
- the proposed approach outperforms all baselines in all tasks for knowledge completion, even when the KG embedding methods are enhanced with the pre-trained product knowledge graph

Task	Link prediction						Product classification			
Relation	complement		co-view		substitute		IsA (*category*)		IsA (*department*)	
Metric	**Hit@10** (a1)	**NDCG@10** (a2)	**Hit@10** (a3)	**NDCG@10** (a4)	**Hit@10** (a5)	**NDCG@10** (a6)	**micro-F1** (a7)	**macro-F1** (a8)	**micro-F1** (a9)	**macro-F1** (a10)
TransE (No PRG)	1.84	1.06	3.27	2.11	12.04	6.56	42.33	69.44	51.72	76.53
TransD (No PRG)	1.97	1.08	3.51	2.24	13.69	6.90	41.82	67.65	50.29	75.45
DistMult (No PRG)	3.47	1.88	6.58	3.41	20.64	9.96	53.75	74.69	61.42	80.73
TransE (With PRG)	3.65	1.82	6.95	3.90	30.22	13.41	45.43	74.93	55.89	81.81
TransH (With PRG)	4.13	1.79	6.88	2.89	30.37	13.56	41.94	64.09	50.12	72.95
TransR (With PRG)	6.06	2.35	8.17	3.43	31.25	14.88	46.37	72.74	53.95	74.11
TransD (With PRG)	4.26	1.95	7.03	2.97	20.71	9.86	50.36	71.02	59.62	82.43
RESCAL (With PRG)	1.64	0.97	1.63	0.87	12.46	5.76	62.89	86.27	72.27	90.97
DistMult (With PRG)	5.69	2.47	9.64	4.05	30.64	12.25	*68.25*	*94.23*	72.09	92.94
ComplEx (With PRG)	*7.81*	*3.36*	*12.38*	*5.77*	*31.25*	*12.60*	67.46	94.02	*72.54*	*97.71*
Our approach	***14.53***	***7.67***	***20.84***	***10.26***	***34.58***	***14.77***	***68.62***	***95.17***	***74.61***	***99.60***

Conclusion

- Product Knowledge Graph:
  - Self-attention-enhanced distributed representation learning model
  - Raw customer activity data
  - Multi-task
  - Complex entity structures
- Future work:
  - Incorporating customer and customer knowledge into PKG [customer-product knowledge graph]
  - Personalized e-commerce services

Summary Discussion

	Customer Interaction Network	Product Knowledge Graph
Strengths	• A method in E-commerce search • Study on properties of CIN • Improvement of performance in E-commerce	• PKG fit the commerce relation better • Lower noise Raw data usable • Multiple feature included
Weakness	• Only tested on Walmart data • Analyzing one graph at a time • Only query and item relation ship & query-query	• Only tested on Walmart data • Multi-task summarization not good • No relationship between query

Thanks!