Task-guided Pair Embedding in Heterogeneous Network

Chanyoung Park[1], Donghyun Kim[2], Qi Zhu[1], Jiawei Han[1], Hwanjo Yu[3]

[1]University of Illinois at Urbana-Champaign

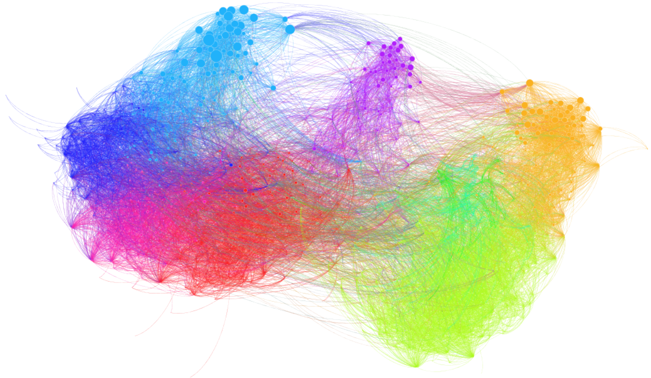
[2]Yahoo! Research

[3]Pohang University of Science and Technology (POSTECH)

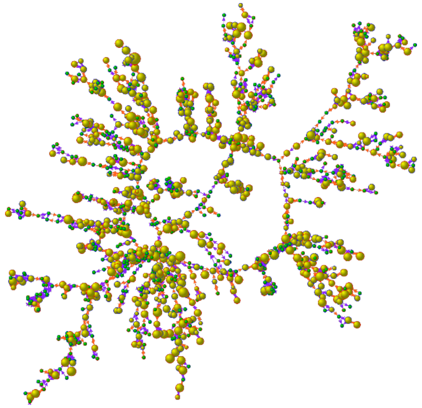
The 28th ACM International Conference on Information and Knowledge Management (CIKM 2019)

Network

- A ubiquitous data structure to model the relationships between entities
- Many types of data can be flexibly formulated as networks

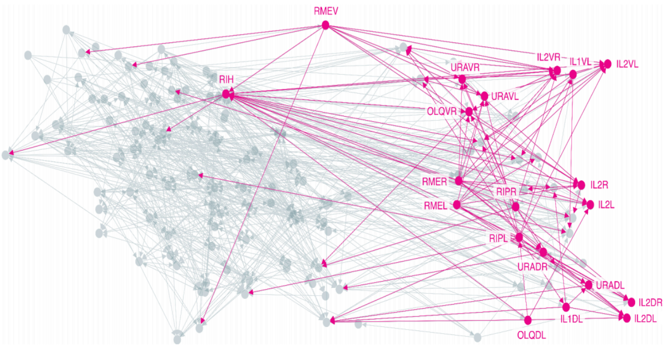


Social Network



Biological Network

Chemical Network

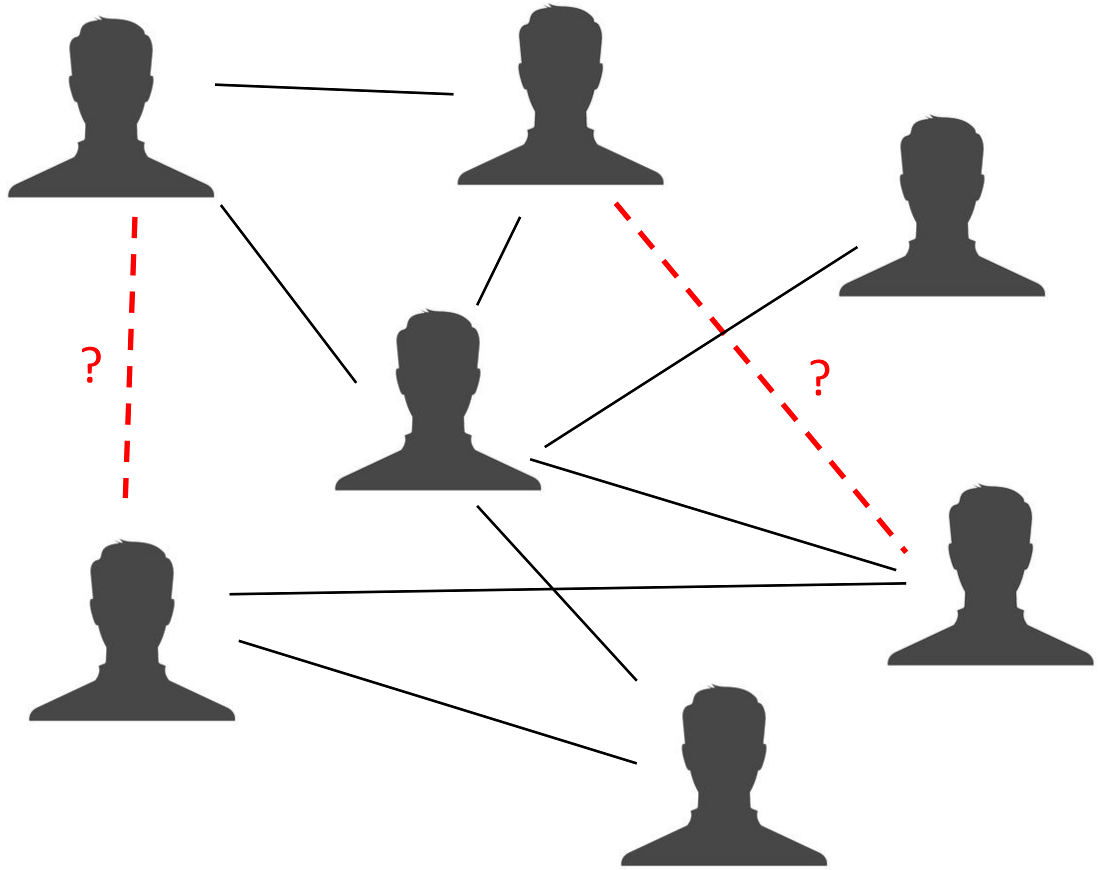


Network of neurons

Classical Tasks in Networks

- Node classification
  - Predict the type of a given node
- Link prediction
  - Predict whether two nodes are linked
- Community detection
  - Identify densely linked clusters of nodes
- Network similarity
  - How similar are two (sub)networks

Example: Link Prediction (Friend Recommendation)

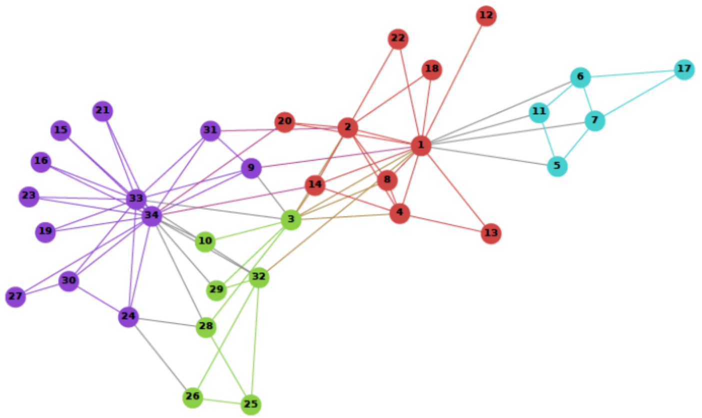


How do we solve these network-related tasks?

→ Node embedding-based methods

Node Embedding

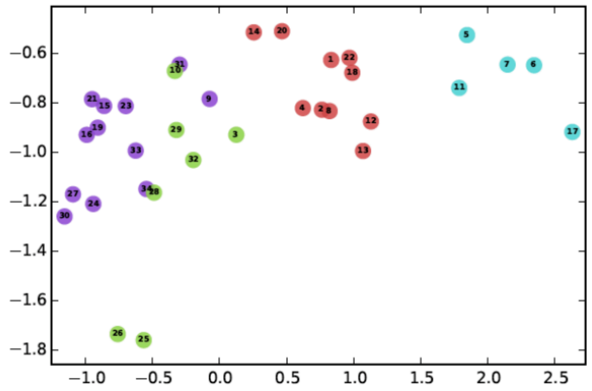
- Find a **low-dimensional vector representation of each node** in a graph while preserving the network structure
  - **Intuition**: Similar nodes in a graph have similar vector representations



Input

Node
embedding method

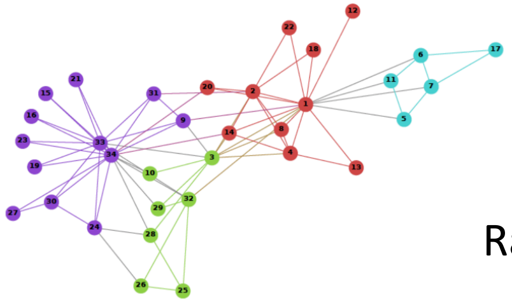
(Deepwalk, node2vec…)



Output

Related Work: Deepwalk (Perozzi et al, 2014)

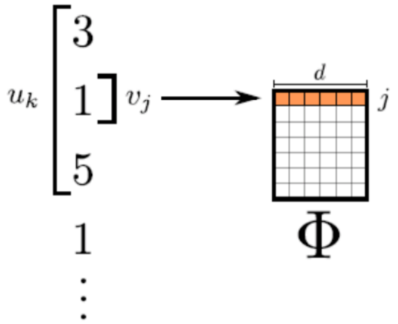
- DeepWalk converts a graph into a collection of node sequences using uniform sampling (truncated random walk)
- **Assuming each sequence as a sentence**, they run the Skip-gram model (Mikolov et al. 2014) to learn representation for each node (like word2vec)



Random walk

$$\mathcal{W}_{v_4} \equiv v_4 \rightarrow v_3 \rightarrow v_1 \rightarrow v_5 \rightarrow v_1 \rightarrow v_{46} \rightarrow v_{51} \rightarrow v_{89}$$

$\mathcal{W}_{v_4} =$



Maximize: $\Pr(v_3|\Phi(v_1))$

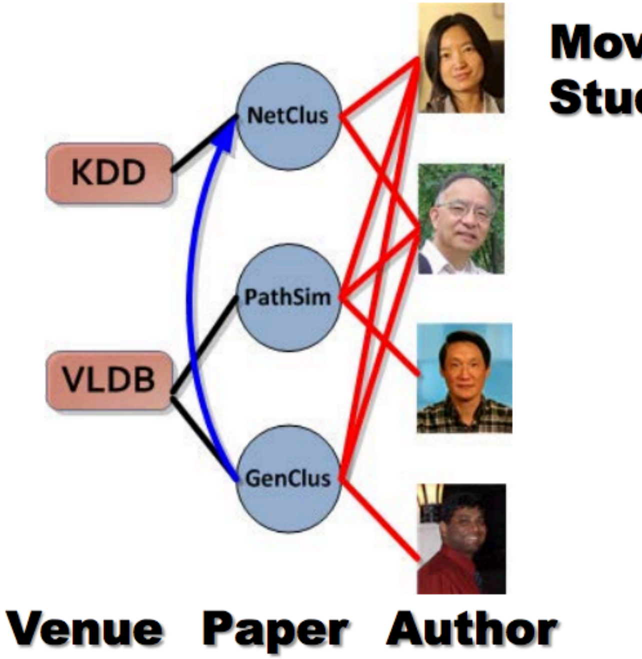
$\Pr(v_5|\Phi(v_1))$

Can only be applied to a network with a single type of nodes and edges.
(not to heterogeneous network)

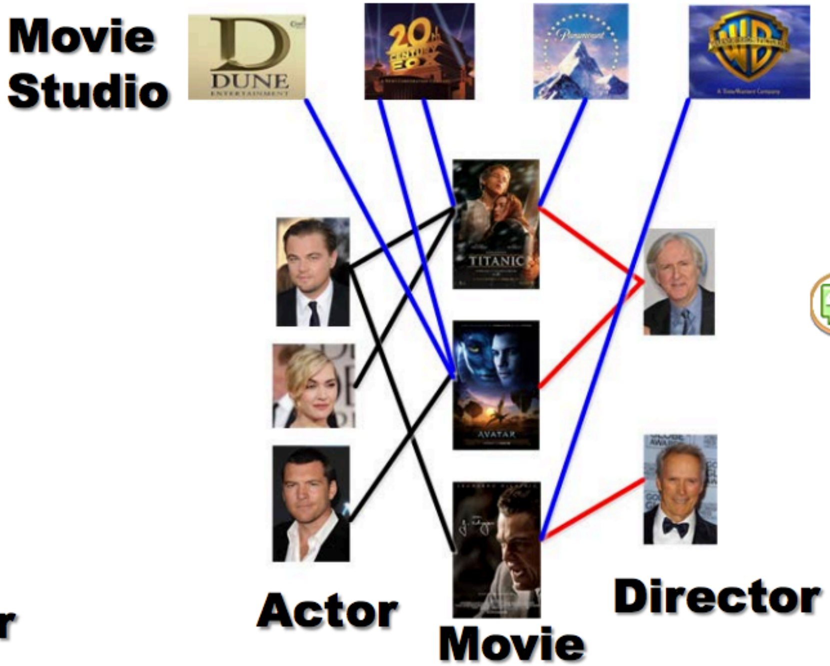
Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014.

Heterogeneous network (HetNet)

- A network with **multiple types of nodes** and **multiple types of edges**
- A lot of networks in reality are heterogeneous network



DBLP Bibliographic Network



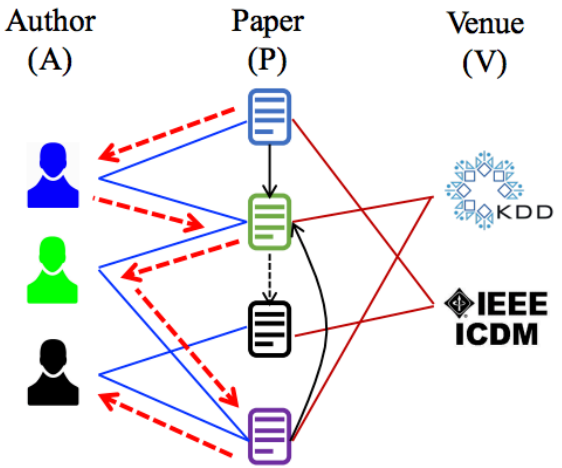
The IMDb Movie Network

The Facebook Network

How do we embed nodes in a heterogeneous network?

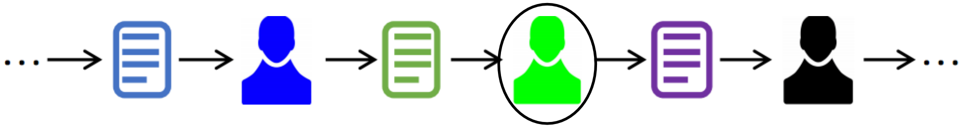
Node Embedding for Heterogeneous Network:
Metapath2vec (Dong et al, 2017)

- Motivation: Deepwalk assumes that each node has a single type → Extend Deepwalk to HetNet!



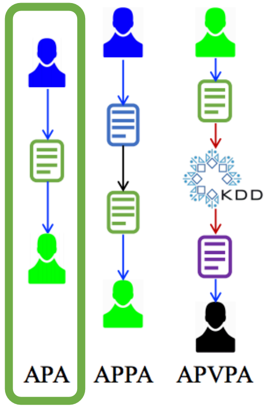
①

Meta-path guided
random walk
(APA)



② Skip-gram model
(like Deepwalk)

Meta-paths

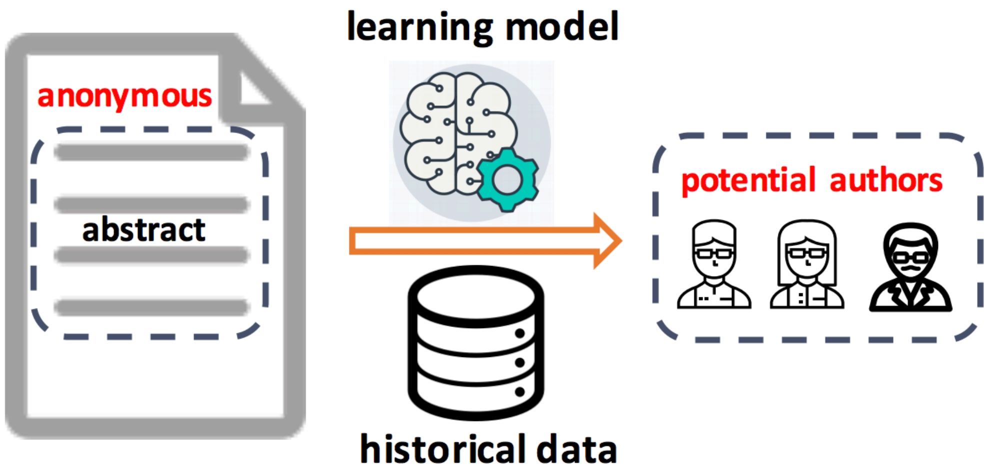


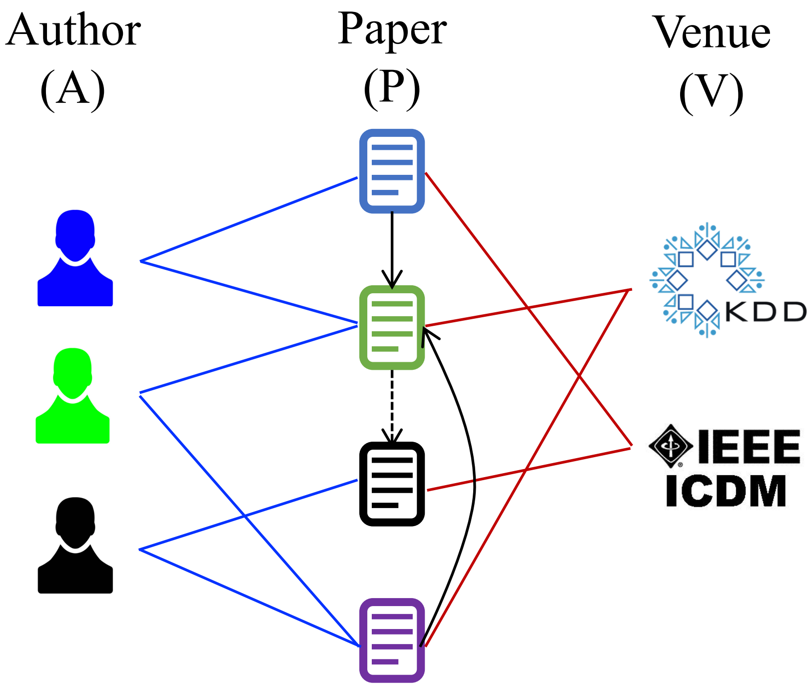
Maximize:

Dong, Yuxiao, Nitesh V. Chawla, and Ananthram Swami. "metapath2vec: Scalable representation learning for heterogeneous networks." *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2017.

Task-guided HetNet embedding

- Instead of learning general node embeddings, what about we focus on a specific task?
- Example: Author Identification
  - Predict the true authors of an anonymized paper given
    - Paper abstract
    - Venue (e.g., KDD, ICDM)
    - References
- Can we predict the true authors? [1,2]



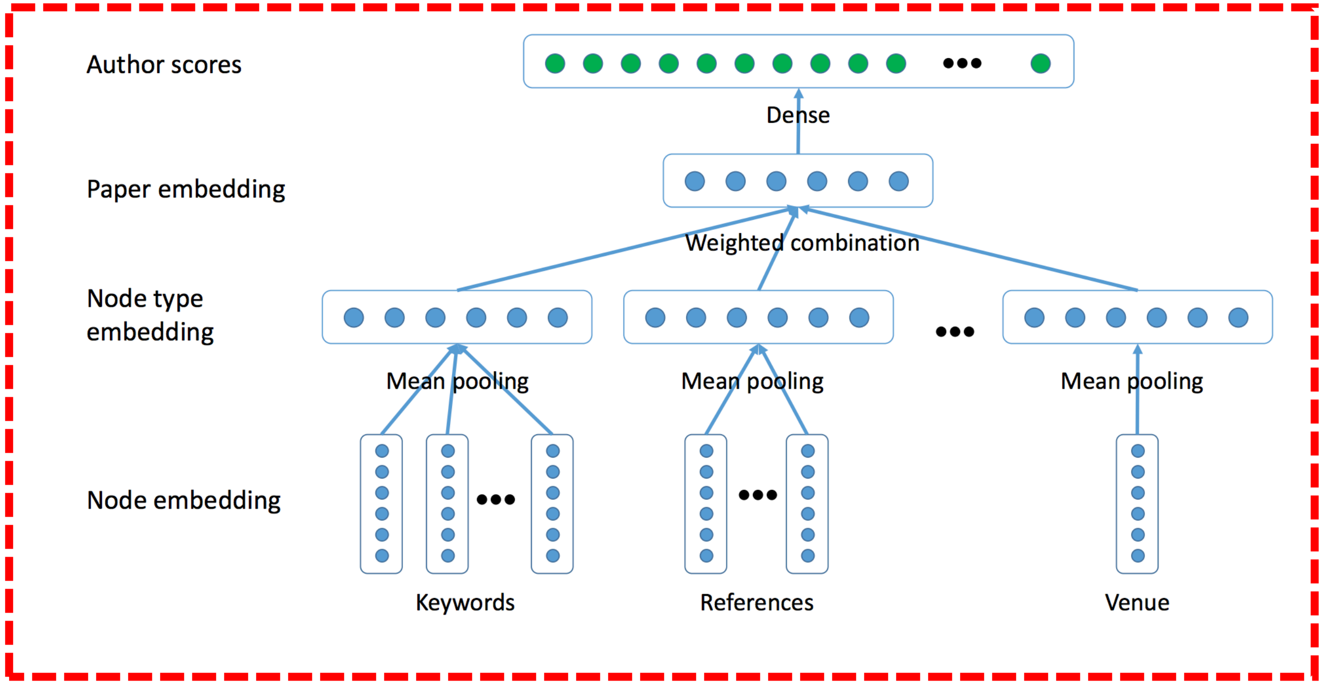


[1] Chen, Ting, and Yizhou Sun. "Task-guided and path-augmented heterogeneous network embedding for author identification." WSDM, 2017.
[2] Zhang, Chuxu, et al. "Camel: Content-Aware and Meta-path Augmented Metric Learning for Author Identification." WWW. 2018.

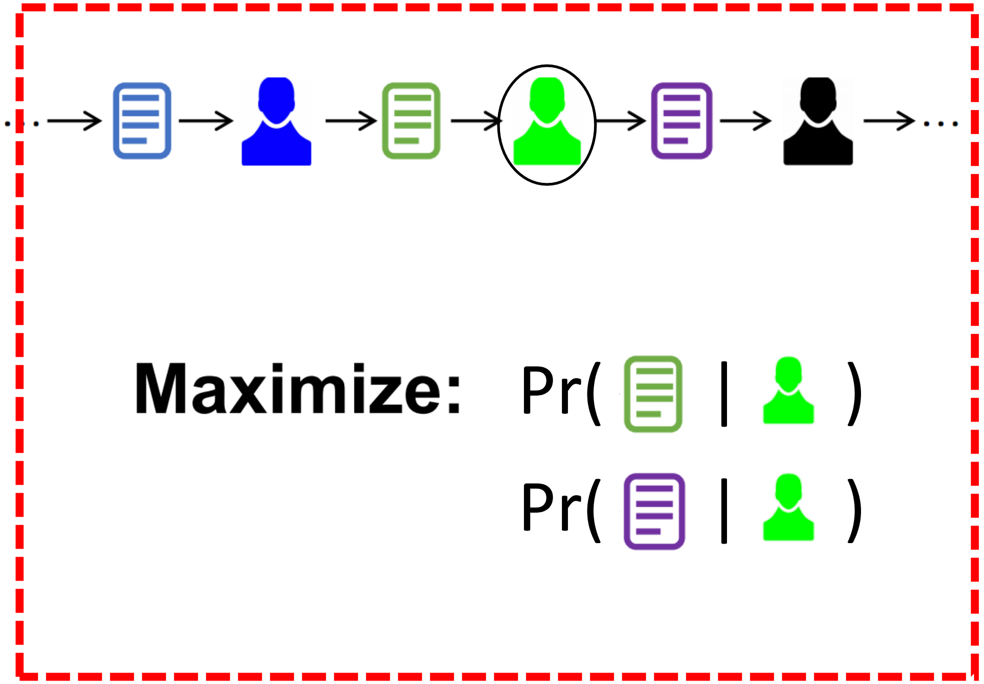
Previous Research on Task-guided HetNet Embedding

[WSDM17] Task-guided and path-augmented heterogeneous network embedding for author identification

- **Step 1**: Combine keywords, venue and references related to a paper to obtain the paper embedding
- **Step 2**: Perform metapath2vec using embeddings learned in step 1



+



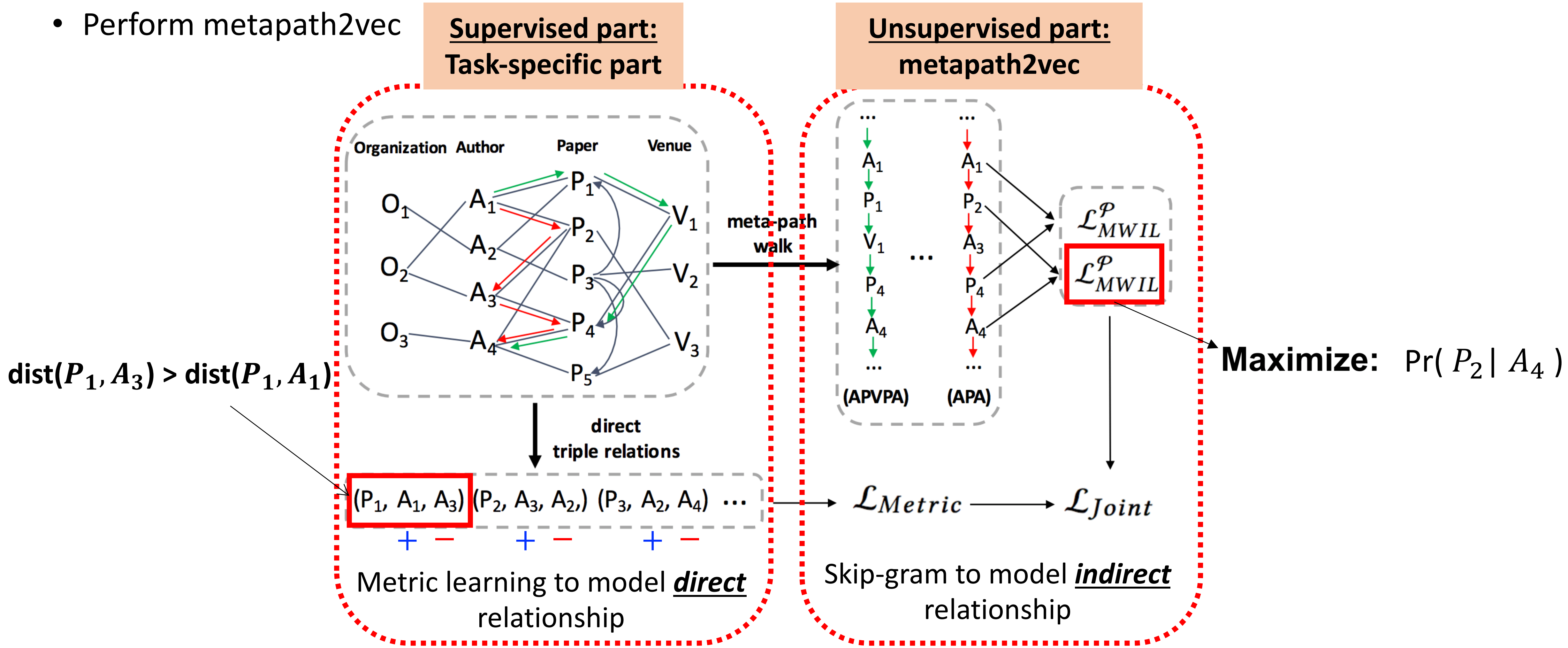
Supervised part:
Task-specific part

Unsupervised part:
metapath2vec

Previous Research on Task-guided Embedding

[WWW18] Camel: Content-Aware and Meta-path Augmented Metric Learning for Author Identification

- Model the paper abstract using a GRU-based encoder
- Perform metapath2vec



Our Motivation

- Directly modeling the **<u>pairwise relationship between two nodes</u>** is crucial for task-guided embedding methods
- The ultimate goal is usually to model the likelihood of the pairwise relationship
  - i.e., Link probability between two nodes
- Example
  - Recommendation
    - The goal is to **model the likelihood of a user favoring an item** (i.e., user–item pairwise relationship)
  - Author identification
    - The goal is to **model the likelihood of a paper being written by an author** (i.e., paper– author pairwise relationship)

- However, previous task-guided embedding methods are **node-centric**

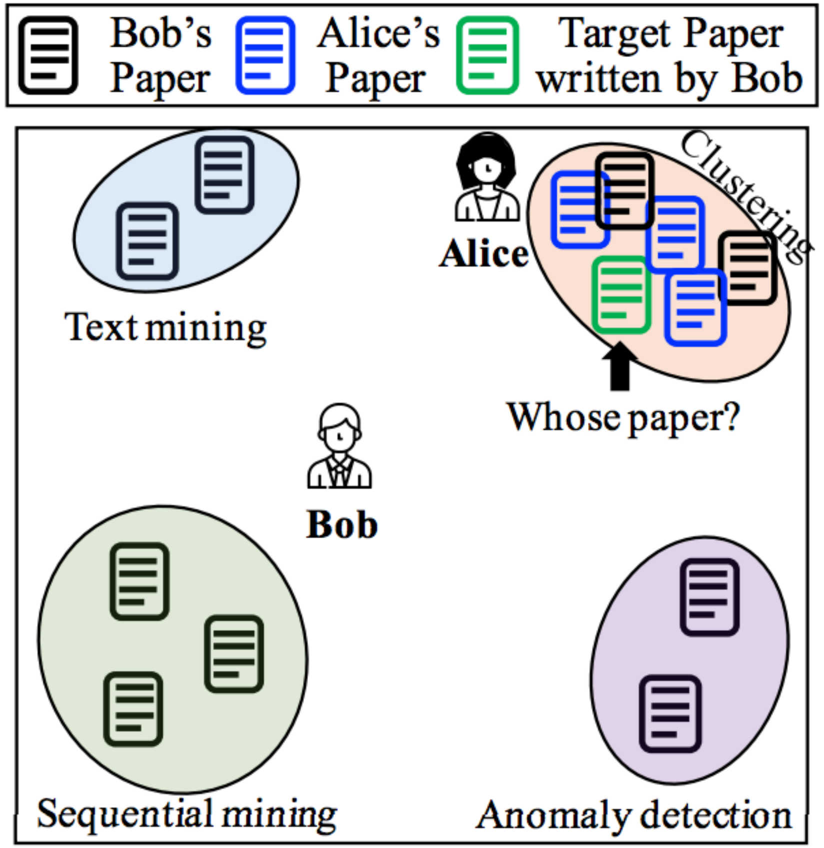
  Step 1. Learn task-guided <u>*node embeddings*</u>

  Step 2. Then, simply use inner product between two node embeddings to compute the pairwise likelihood

We devise **pair embedding** to directly model the pairwise relationship

Toy example: Author identification (Node embedding)

- Assumption
  - Bob has written multiple papers in various research areas
  - Alice only worked on “Clustering” topic
- Case 1) Node embedding
- Should find **a single optimal point** to satisfy all relationship
  - **Bob’s embedding**: Should satisfy his relationship with various research areas
  - **Alice’s embedding**: Should be close to papers whose topics are “clustering”
- **Question**: What about a new paper on “Clustering” written by Bob?
  - It will be embedded together with “Clustering” papers, and therefore close to Alice

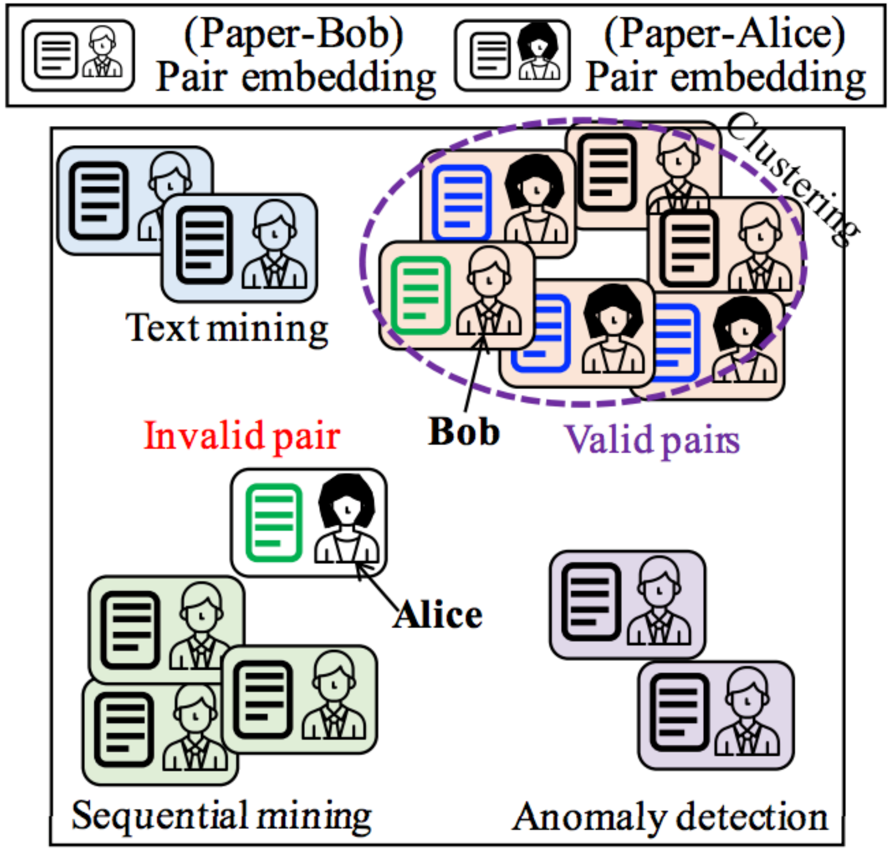


(a) Node embedding

Our approach: Pair Embedding

- Assumption
  - Bob has written multiple papers in various research areas
  - Alice only worked on "Clustering" topic
- Case 2: Pair embedding
- Embed each paper–author pair such that each pair embedding independently captures …
  1. Associated research topic
  2. Pair validity information
     - Whether the pair is valid or not

     = Whether the paper is written by the author within a pair
- By doing so, we want the **pairs to be embedded close to each other if both of the above two conditions hold**

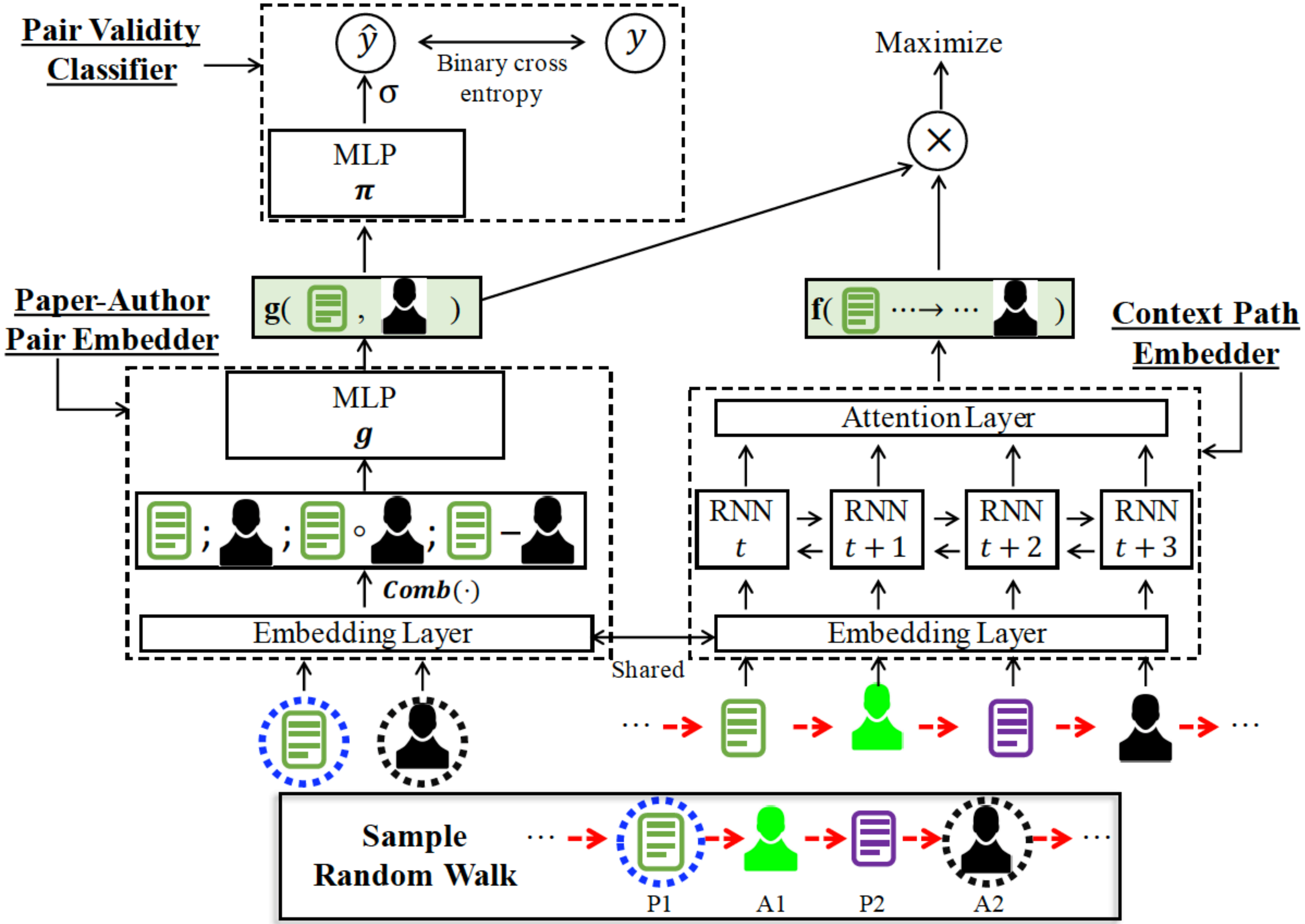


(b) Pair embedding

Summary: Our goals

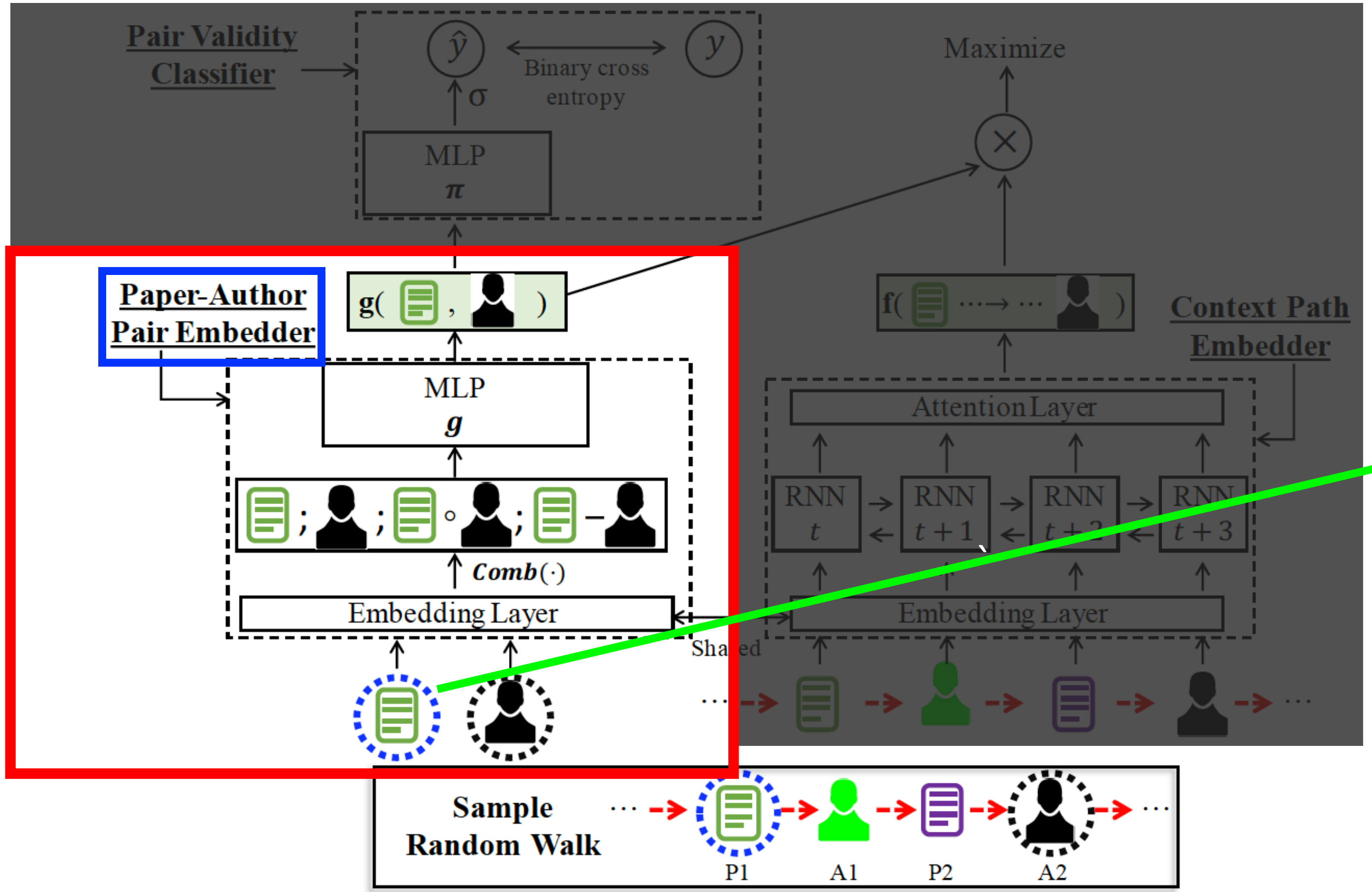
1. To model the **semantics** (e.g., research topic) behind the pairwise relationship
2. To model the **validity** of the pair regarding a specific task
    - This work: Author identification
        - Given a paper–author pair, whether the paper in the pair is written by the author in the pair

Proposed Method: TapEm
Overall Architecture

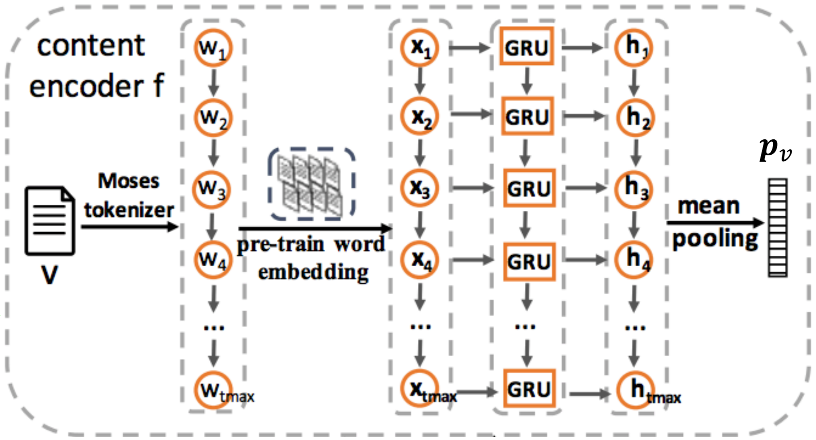


Proposed Method: TaPEm

- **1) Context Path-aware Pair Embedder**
  - Step 1: Pair Embedder (Embedding Paper–Author Pair)

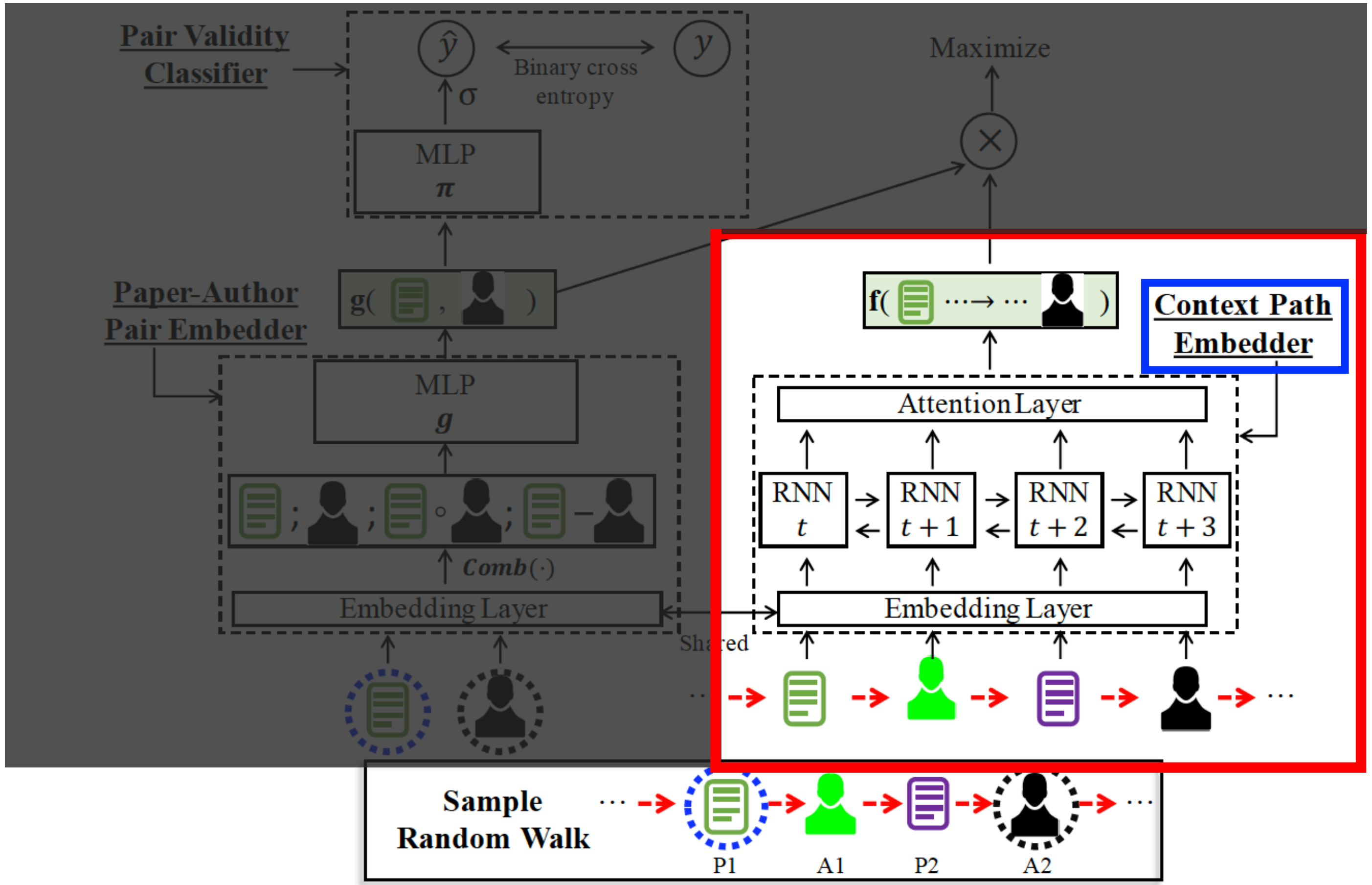


$$\mathbf{p}_v = \text{PaperEncoder}(O_v)$$



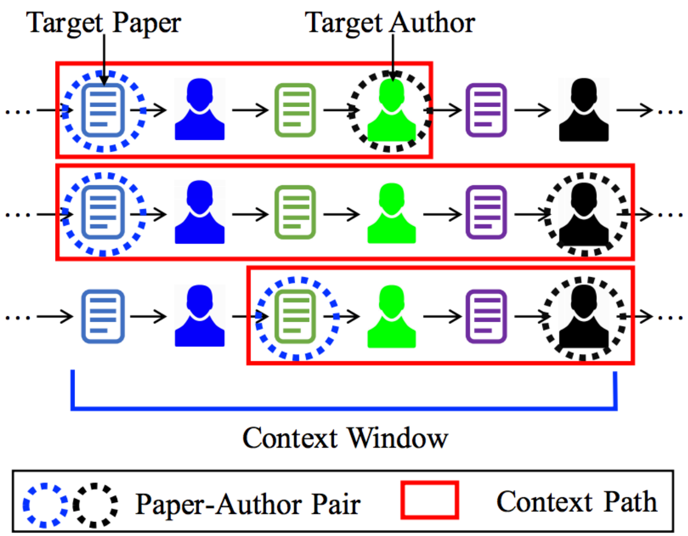
Proposed Method: TaPEm

- **1) Context Path-aware Pair Embedder**
  - Step 2: Context Path Embedder (Embedding Context Path)



What is a **context path**?

A sequence of nodes between a target node pair

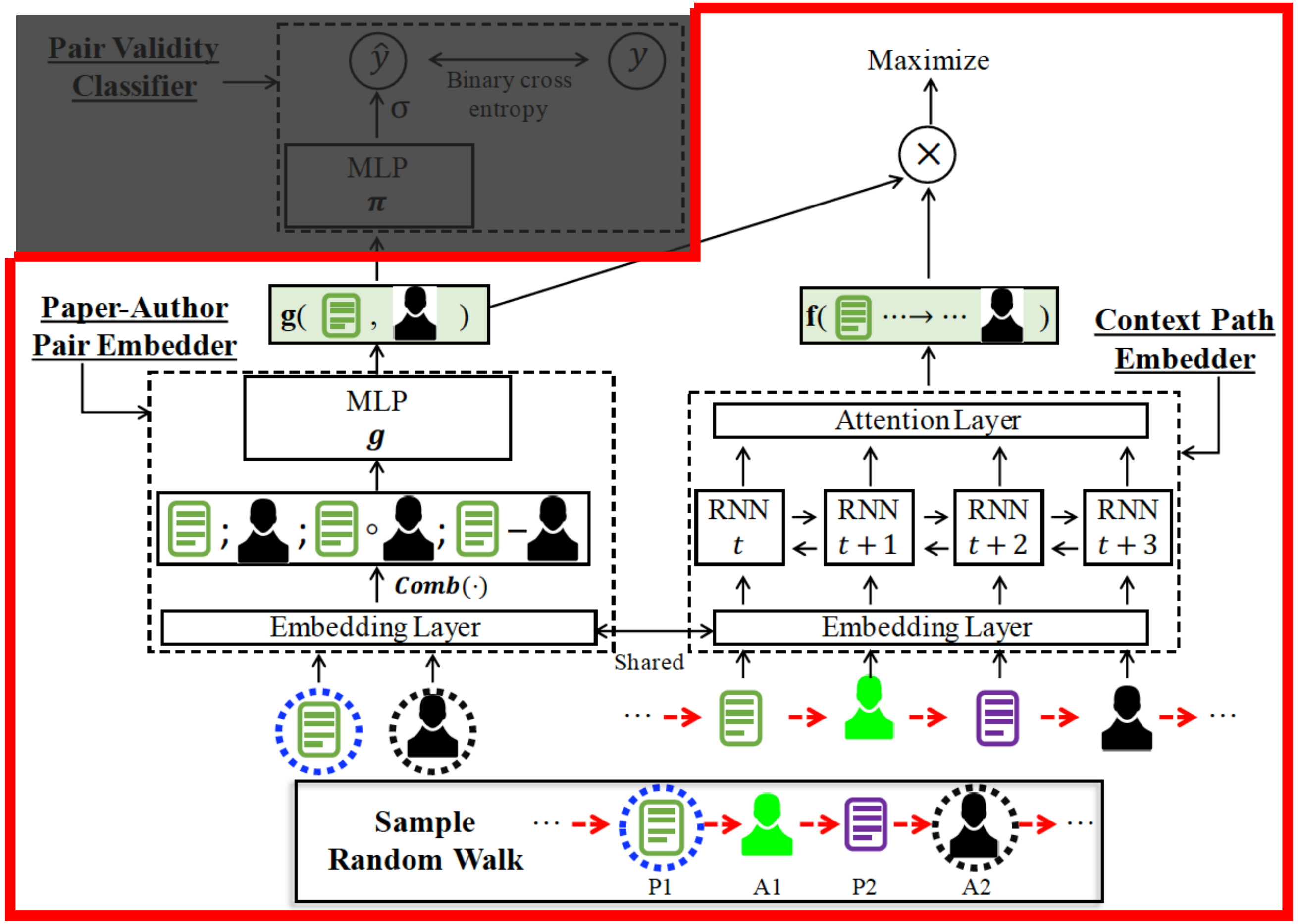


Why do we consider the **context path**?

We can infer the research topic related to the pair (v, u) by examining the path between paper v and author u

Proposed Method: TaPEm

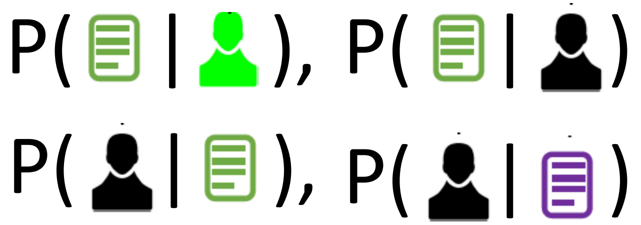
- **1) Context Path-aware Pair Embedder**
  - Step 3: Injecting Context Information into Pairs



Objective (Pair embedding)

Predict pair using its context path

~~Skip-Gram~~



$$\mathcal{L}_{\text{ctx}}(v,u) = \sum_{c \in C^{\mathcal{P}}_{v \to u}} -\log p((v,u)|c,\mathcal{P})$$

$$p((v,u)|c,\mathcal{P}) = \frac{\exp\left[(\mathbf{g}(v,u) \cdot \mathbf{f}(c))\right]}{\sum_{c' \in C^{\mathcal{P}}_{*}} \exp\left[(\mathbf{g}(v,u) \cdot \mathbf{f}(c'))\right]}$$

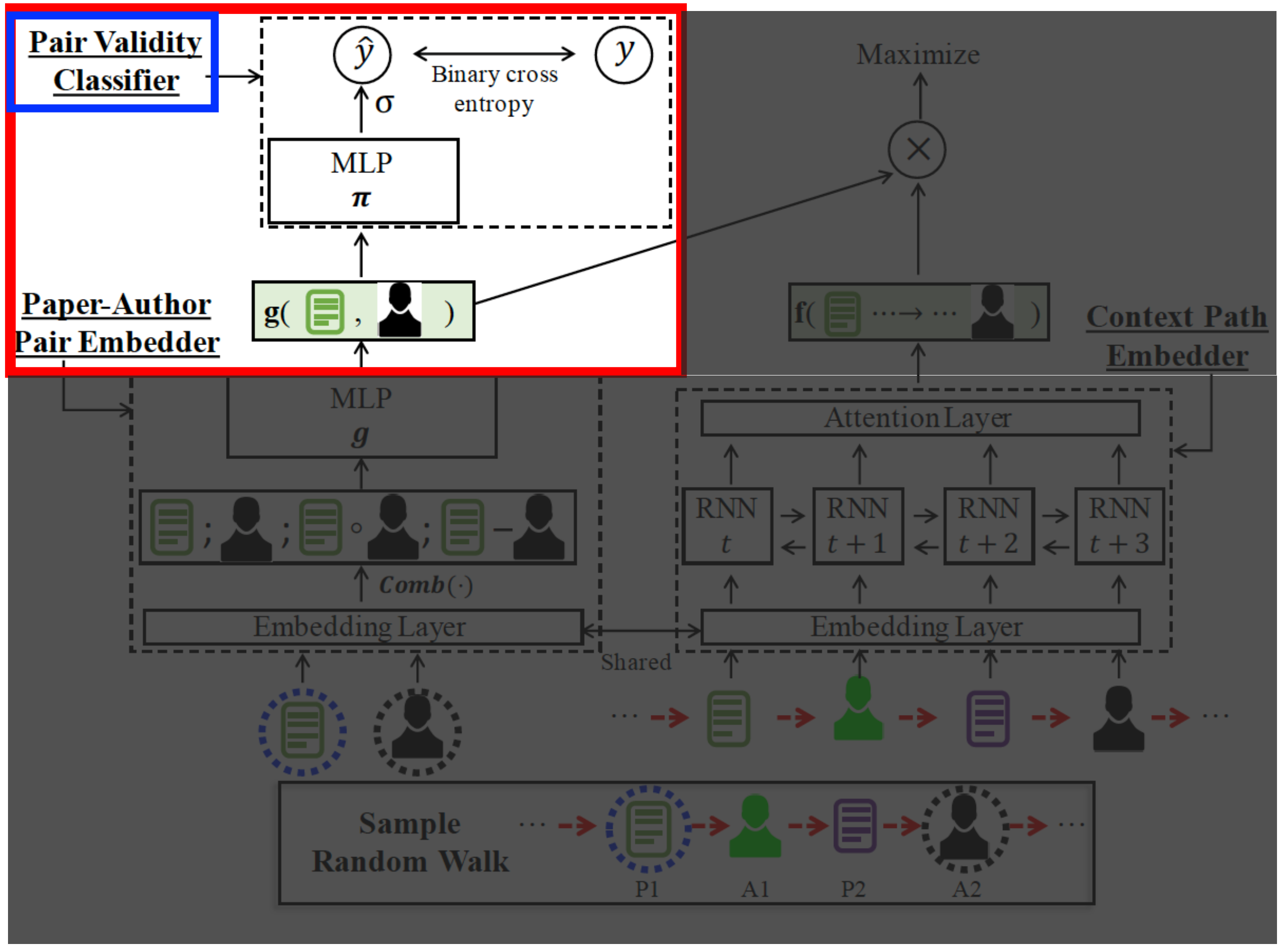
Benefit

Pair embedding ≈ Embeddings of frequent context paths

→ Pair embedding encodes its related research topic

Proposed Method: TaPEm

- **2) Pair Validity Classifier** (Validity of Pair Embedding)



Objective

- Classify whether the pair is valid or not

$$\mathcal{L}_{\mathrm{pv}}(v,u) = y_{v,u}\,\sigma(\boldsymbol{\pi}(\mathbf{g}(v,u))) + (1-y_{v,u})(1-\sigma(\boldsymbol{\pi}(\mathbf{g}(v,u))))$$

$$y_{v,u} = \begin{cases} 1, & \text{paper } v \text{ is written by author } u \\ 0, & \text{paper } v \text{ is not written by author } u \end{cases}$$

Benefit

- Enables to identify **relatively less active authors**
  - The training of the embedding is no longer solely based on the frequency (Limitation of Skip-Gram)
- Two nodes will be embedded close to each other if
  1. Related to a similar research topic
  2. **The pair itself is valid**

Joint Objective

Context Path-aware Pair Embedder

Pair Validity Classifier

$$\mathcal{L} = \sum_{\mathcal{P} \in \mathcal{S}(\mathcal{P})} \sum_{\mathbf{w} \in \mathcal{W}_{\mathcal{P}}} \sum_{v \in \mathbf{w}} \sum_{u \in \mathbf{w}[C_v - \tau : C_v + \tau]} \left[\boxed{\mathcal{L}_{\mathrm{ctx}}(v, u)} + \boxed{\mathcal{L}_{\mathrm{pv}}(v, u)} \right]$$

- $S(P)$: a set of meta-path scheme
- W_p: a set of random walks guided by meta-path p
- τ: window size
- C_v: position of paper v in walk w

Experiments

- Dataset: AMiner dataset
  - Extracted 10 years of data (2006 ~ 2015)
  - Removed the papers published in venues with limited publications
  - Two versions
    - AMiner-Top: Selected 18 top conferences from AI, DM, DB, IS, CV, and CL
    - AMiner-Full: All venues

Statistics	AMiner-Top	AMiner-Full
# authors	27,920	536,811
# papers	21,808	447,289
# venues	18	389

AI: ICML, AAAI, IJCAI. **DM**: KDD, WSDM, ICDM. **DB**: SIGMOD, VLDB, ICDE.
IS: WWW, SIGIR, CIKM. **CV**: CVPR, ICCV, ECCV. **CL**: ACL, EMNLP, NAACL

Experiments

- Baselines
  1. **Feature engineering–based** supervised method
  2. **General purpose** heterogeneous network embedding method
     - Metapath2vec [KDD17] (Dong et al, 2017)
  3. **Task-guided** heterogeneous network embedding methods
     - HNE [WSDM17] (Chen et al, 2017)
     - Camel [WWW18] (Zhang et al, 2018)
     - TaPEm$_{\mathrm{npv}}$: TaPEm without pair validity classifier

Experiments: All authors (Active + Inactive)

Dataset		Metric	Sup	MPV	HNE	Camel	TaPEm$_{npv}$	TaPEm	Impr.
AMiner-Top	T=2013	Rec@5	0.5460	0.5274	0.4874	0.5902	0.6405	**0.6807**	15.33%
		Rec@10	0.6227	0.6746	0.6301	0.7370	0.7677	**0.7849**	6.50%
		Prec@5	0.2285	0.2148	0.2051	0.2439	0.2662	**0.2835**	16.24%
		Prec@10	0.1323	0.1401	0.1334	0.1555	0.1632	**0.1664**	7.01%
		F1@5	0.3222	0.3052	0.2888	0.3452	0.3761	**0.4003**	15.96%
		F1@10	0.2182	0.2320	0.2202	0.2568	0.2691	**0.2746**	6.93%
		AUC	0.7817	0.8887	0.8614	0.9112	0.9164	**0.9178**	0.72%
	T=2014	Rec@5	0.5142	0.5116	0.4665	0.5625	0.6121	**0.6577**	16.92%
		Rec@10	0.5792	0.6661	0.6185	0.7198	0.7471	**0.7698**	6.95%
		Prec@5	0.2508	0.2457	0.2284	0.2706	0.2962	**0.3148**	16.33%
		Prec@10	0.1447	0.1636	0.1538	0.1776	0.1851	**0.1898**	6.87%
		F1@5	0.3371	0.3320	0.3066	0.3654	0.3992	**0.4258**	16.53%
		F1@10	0.2316	0.2627	0.2463	0.2849	0.2967	**0.3045**	6.88%
		AUC	0.7359	0.8904	0.8619	0.9087	0.9112	**0.9206**	1.31%

Dataset		Metric	Sup	MPV	HNE	Camel	TaPEm$_{npv}$	TaPEm	Impr.
AMiner-All	T=2013	Rec@5	0.6096	0.5990	0.6110	0.5458	0.7049	**0.7097**	16.15%
		Rec@10	0.6409	0.7317	0.7166	0.6811	0.8121	**0.8237**	12.57%
		Prec@5	0.2679	0.2562	0.2679	0.2393	0.3076	**0.3087**	15.23%
		Prec@10	0.1418	0.1595	0.1590	0.1508	0.1795	**0.1818**	13.98%
		F1@5	0.3722	0.3589	0.3724	0.3327	0.4283	**0.4303**	15.55%
		F1@10	0.2322	0.2619	0.2602	0.2470	0.2940	**0.2978**	13.71%
		AUC	0.7641	0.8923	0.8855	0.8768	0.9291	**0.9337**	4.64%
	T=2014	Rec@5	0.6203	0.5768	0.5842	0.5494	0.6742	**0.6840**	10.27%
		Rec@10	0.6570	0.7114	0.6927	0.6835	0.7952	**0.7998**	12.43%
		Prec@5	0.2825	0.2586	0.2689	0.2529	0.3068	**0.3109**	10.05%
		Prec@10	0.1510	0.1623	0.1611	0.1588	0.1840	**0.1850**	13.99%
		F1@5	0.3882	0.3571	0.3683	0.3464	0.4217	**0.4275**	10.12%
		F1@10	0.2455	0.2643	0.2614	0.2577	0.2989	**0.3005**	13.70%
		AUC	0.7829	0.8834	0.8747	0.8770	0.9243	**0.9245**	4.65%

- **TaPEm >> Rest (especially when N is small)**
  - TapEm captures the fine-grained pairwise relationship between two nodes
    - **Pushes true authors to the top ranks**
- **TaPEm$_{npv}$ > Rest**
  - Pair embedding framework > Skip-gram
- **TapEm > TaPEm$_{npv}$**
  - Pair validity classifier encodes pair validity information into the pair embedding

Experiments: Inactive Authors

- The skip-gram based model is **biased to active authors**
  - Most authors publish only few papers
    - 92% of authors in AMiner dataset published less than 6 publications
  - Inactive authors: Authors with less than 6 publications

	T	Methods	Recall@N $N=1$	Recall@N $N=2$	Recall@N $N=5$	Recall@N $N=10$	Precision@N $N=1$	Precision@N $N=2$	Precision@N $N=5$	Precision@N $N=10$	F1@N $N=1$	F1@N $N=2$	F1@N $N=5$	F1@N $N=10$	AUC
AMiner-Top	2013	Camel	0.1808	0.3035	0.5012	0.6646	0.3155	0.2734	0.1887	0.1244	0.2299	0.2877	0.2742	0.2096	0.8854
		TaPEm	**0.2677**	**0.4131**	**0.6037**	**0.7220**	**0.4496**	**0.3697**	**0.2251**	**0.1360**	**0.3356**	**0.3902**	**0.3279**	**0.2289**	**0.8935**
		Improve.	48.06%	36.11%	20.45%	8.64%	42.50%	35.22%	19.29%	9.32%	45.98%	35.63%	19.58%	9.21%	0.91%
	2014	Camel	0.1624	0.2739	0.4831	0.6619	0.3372	0.2865	0.2094	0.1440	0.2192	0.2801	0.2922	0.2365	0.8909
		TaPEm	**0.2312**	**0.3670**	**0.5679**	**0.6900**	**0.4515**	**0.3759**	**0.2433**	**0.1507**	**0.3058**	**0.3714**	**0.3406**	**0.2473**	**0.8934**
		Improve.	42.36%	33.99%	17.55%	4.25%	33.90%	31.20%	16.19%	4.65%	39.51%	32.60%	16.56%	4.57%	0.28%

- TapEm performs much better on inactive authors
  - Benefit of **pair embedding + pair validity classifier**

Experiments: Case Study

Target paper

Target author

…P → A → P → A → P → A → P → A…

Context-path

- Case studies to see how TapEm ranks active authors
  - Two author groups exist within a context path
    - 1) True authors, 2) Frequently appearing false authors

- **Case 1: True authors contain an active author**

Paper: (CIKM'06) Mining compressed commodity workflows from massive RFID datasets			
	Author (num. publications)	Rank Camel	Rank TaPEm
True authors	**Jiawei Han** (141)	1	8
	Xiaolei Li (12)	198	1
	Hector Gonzalez (9)	296	81
Frequently appearing false authors	Yizhou Sun (23)	94	418
	Jae-Gil Lee (10)	323	196
	John Paul Sondag (1)	1043	3650

- **Case 3: Both author groups contain an active author**

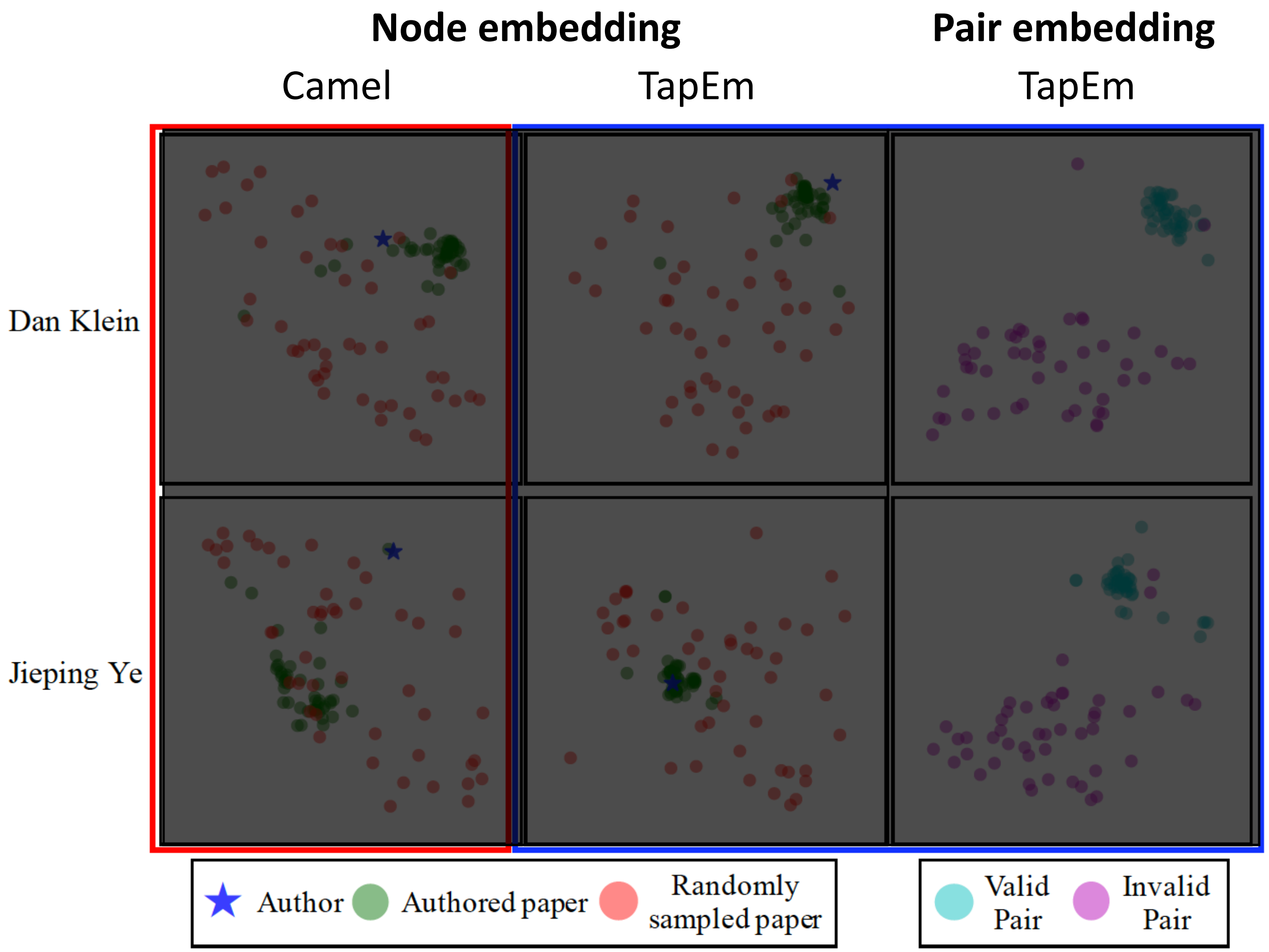
Paper: (KDD'06) Generating semantic annotations for frequent patterns with context analysis			
	Author (num. publications)	Rank Camel	Rank TaPEm
True authors	**Jiawei Han** (141)	1	14
	Qiaozhu Mei (21)	44	9
	Dong Xin (20)	130	26
Frequently appearing false authors	**Philip S.Yu** (122)	7	41
	Xifeng Yan (36)	15	19
	Charu C.Aggarwal (30)	16	303

- **Case 2: Frequently appearing authors contain an active author**

Paper: (KDD'06) A mixture model for contextual text mining			
	Author (num. publications)	Rank Camel	Rank TaPEm
True authors	Cheng Xiang Zhai (51)	4	3
	Qiaozhu Mei (21)	24	4
Frequently appearing false authors	**Jiawei Han** (141)	2	122
	Yintao Yu (6)	601	372

- Camel simply ranks active authors to high ranks (due to Skip-gram)
- TapEm is robust to the activeness of authors (due to pair embedding framework)

Experiments: Visualization of the embeddings



- Node embedding of TapEm
  1. More **tightly grouped together**
  2. The author embedding is closer to the cluster of the authored papers

TaPEm generates **more accurate** embeddings for paper and author

- Pair embedding of TapEm
  - Makes it **even easier to distinguish whether a pair is valid or not**

Pair embedding is useful for task-guided heterogeneous network embedding

Conclusion

- Proposed the pair embedding framework for heterogeneous network
  - Useful for tasks whose goal is to **predict the likelihood of pairwise relationship between two nodes**
- Directly focused on the **pairwise relationship** between two nodes
  - Learn the **pair embedding** instead of node embedding
- The pair validity classifier is effective in **identifying less active true authors**