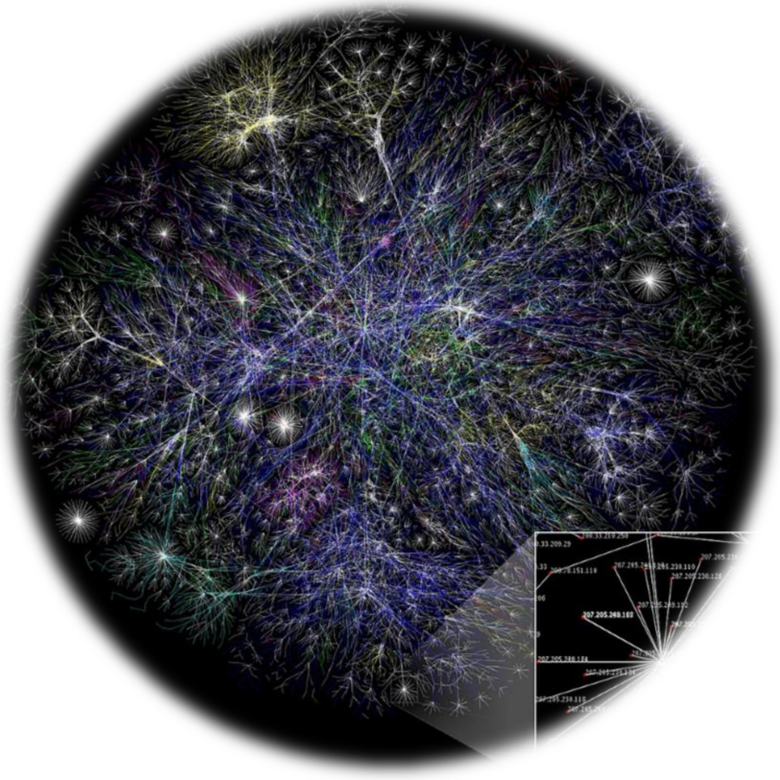
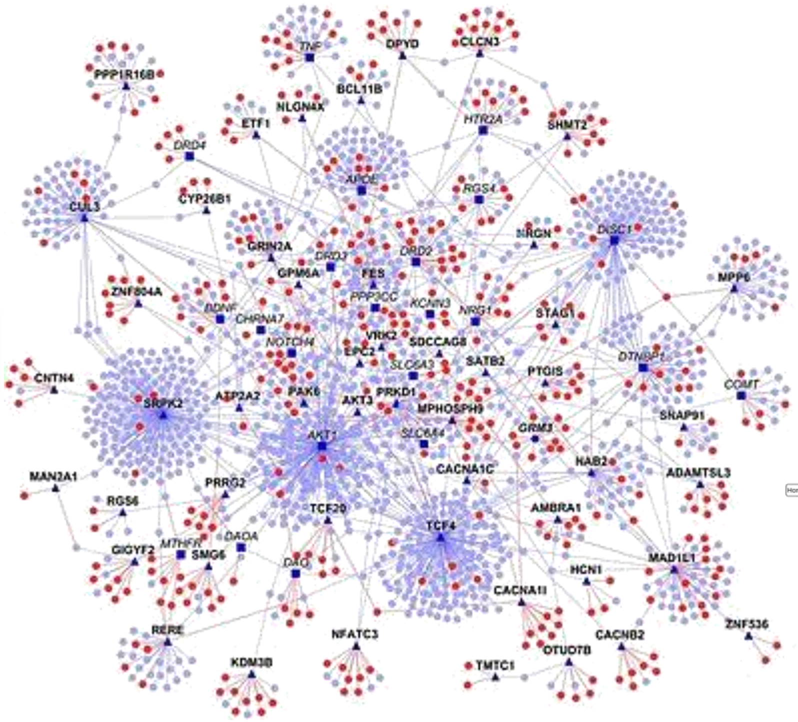
Deep Learning on Graphs

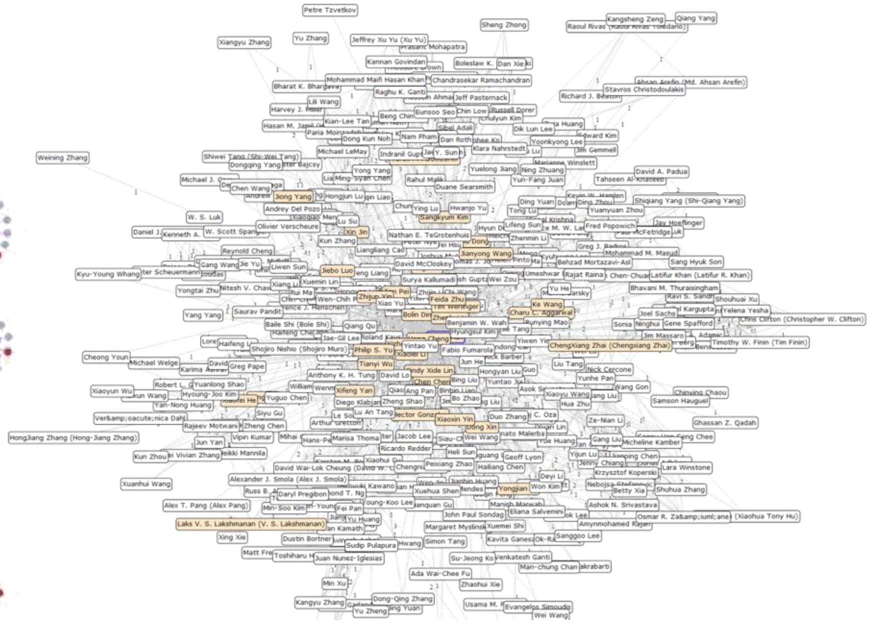
Prof. Kuan-Ting Lai
National Taipei University of Technology
2019/11/27

Graphs (Networks)

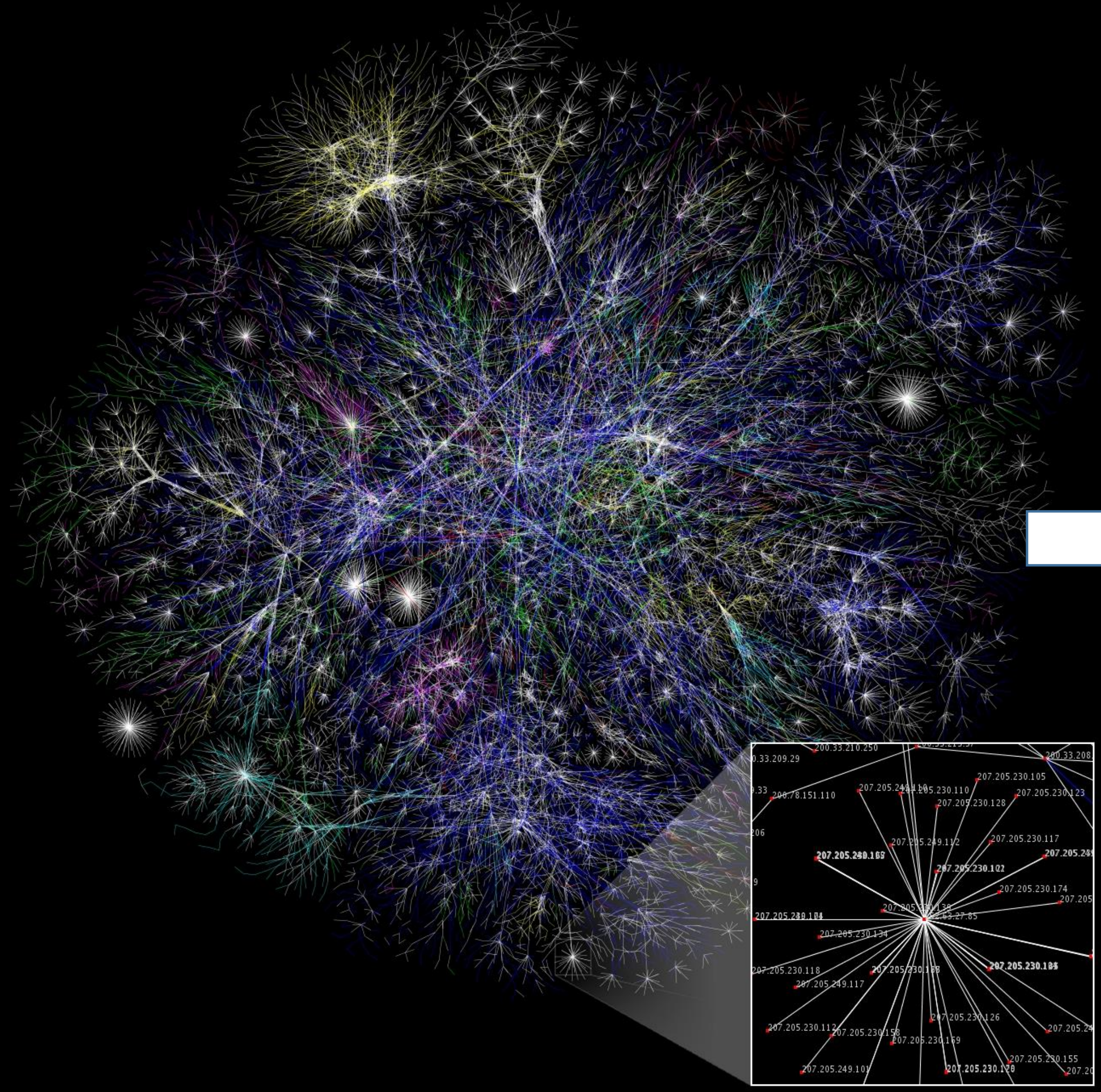
- Ubiquitous in our life
  - Ex: the Internet, Social Networks, Protein-interaction

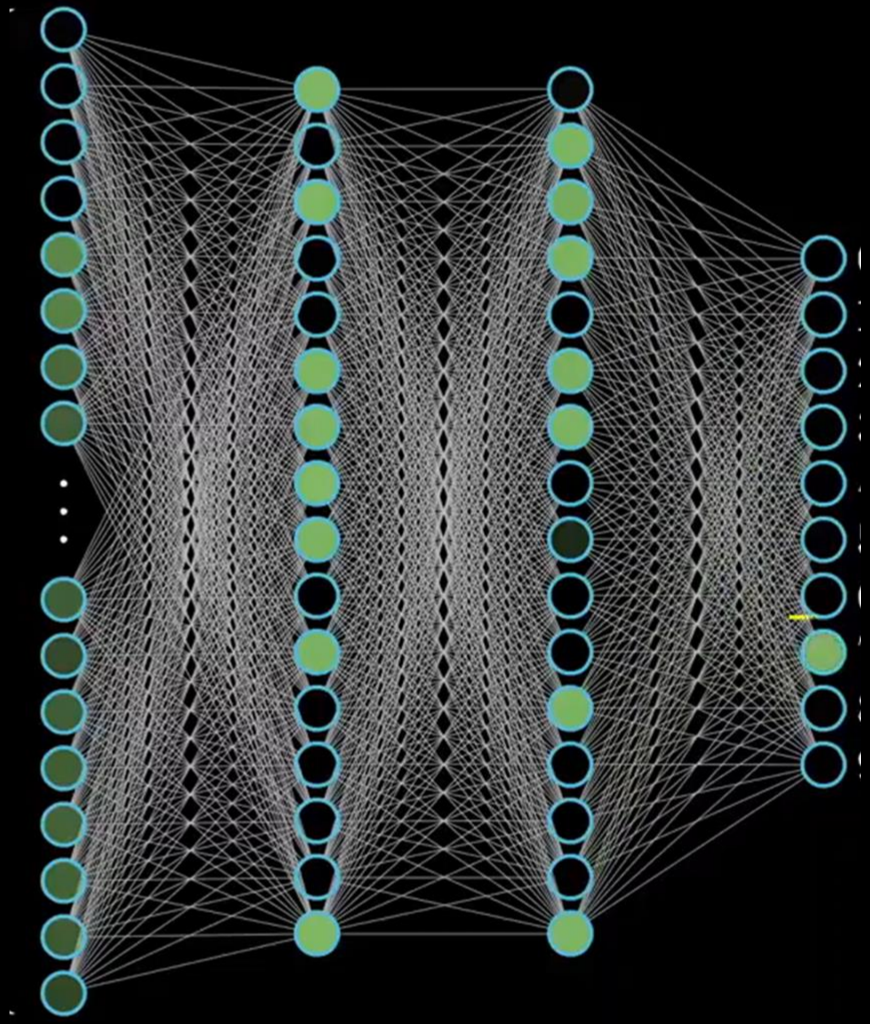






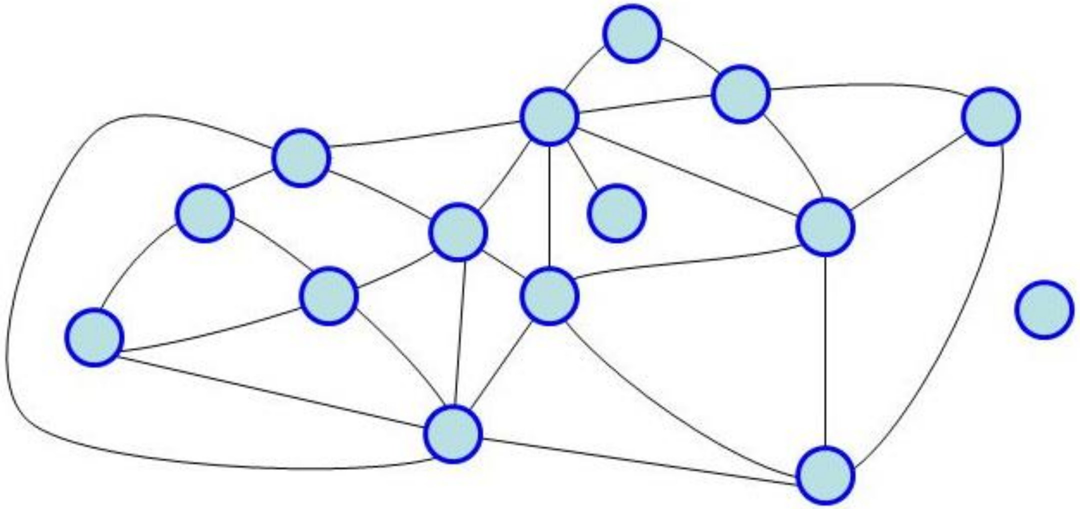
Graph + Deep Learning





Graph Terminology

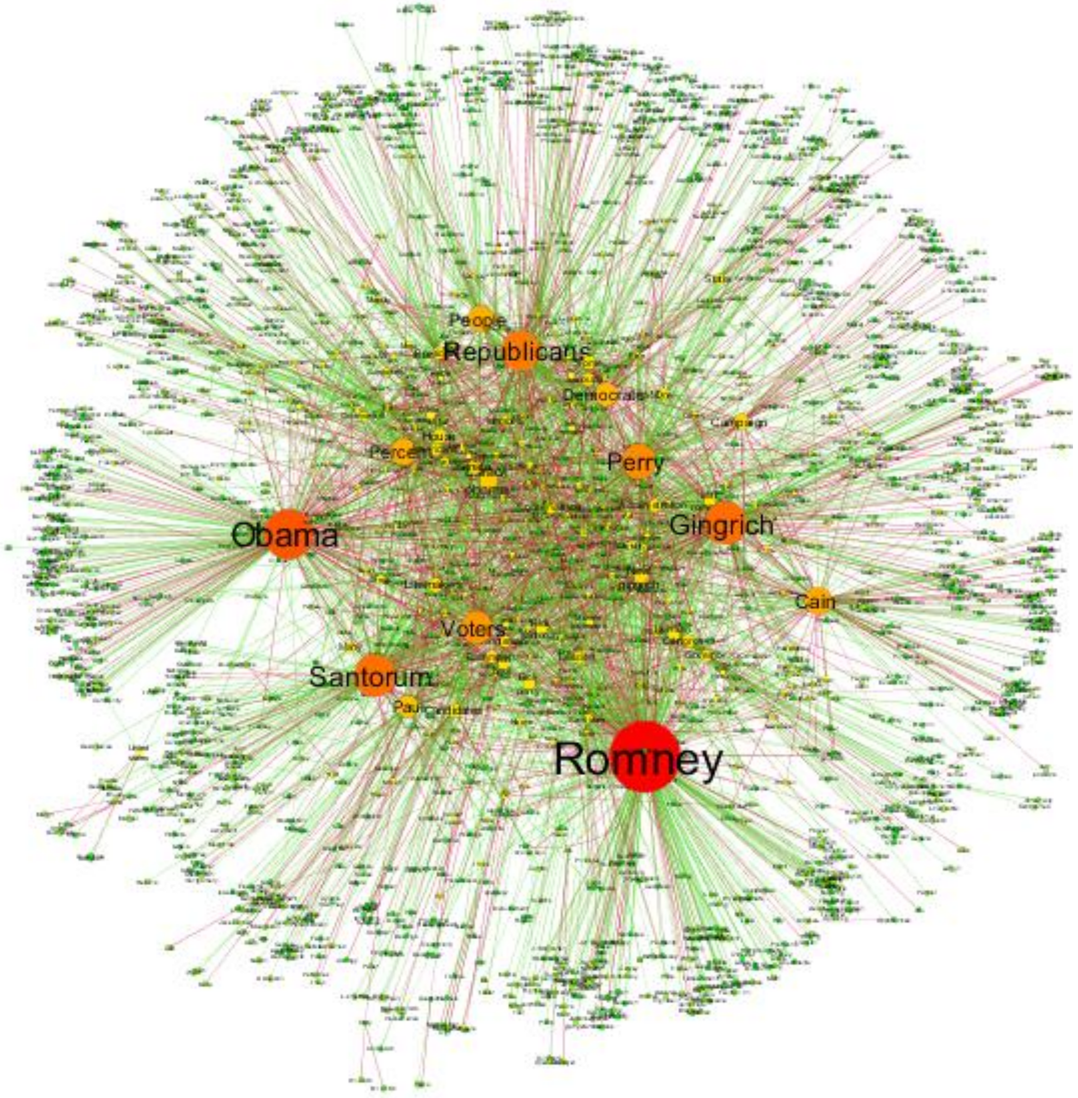
- An *edge* (*link*) connects two *vertices* (*nodes*)
- Two vertices are *adjacent* if they are *connected*
- An edge is *incident* with the two vertices it connects
- The *degree* of a vertex is the number of incident edges



Marshall Shepherd, https://slideplayer.com/slide/7806012/

Network Analysis

- Vertex importance
- Role discovery
- Information propagation
- Link prediction
- Community detection
- Recommender System



Deep Learning on Graphs

- Graph Recurrent Neural Networks
- Graph Convolutional Networks (GCNs)
- Graph Autoencoders (GAEs)
- Graph Reinforcement Learning
- Graph Adversarial Methods

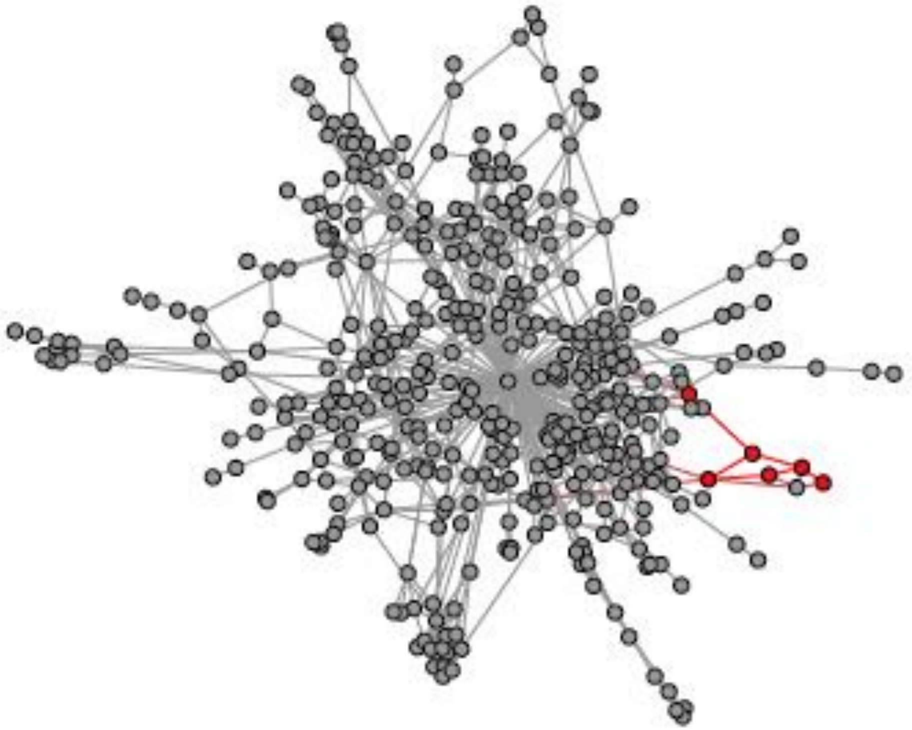
Zhang *et al.*, “Deep Learning on Graphs: A Survey,” 2018

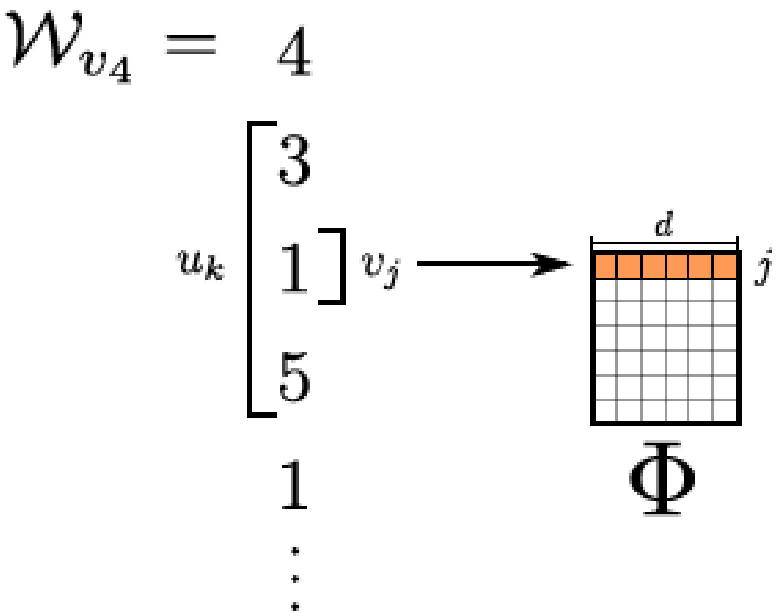
Learning Vertex Features

- Graph Embedding (Random walk + Word embedding)
  - DeepWalk (2014), LINE (2015), node2vec (2016), DRNE (2018),...
- Graph Convolutional Networks (GCNs)
  - Bruna et al. (2014), Atwood & Towsley (2016), Niepert et al. (2016), Defferrard et al. (2016), Kipf & Welling (2017),...

DeepWalk (2014)

- Random Walk + Word Embedding





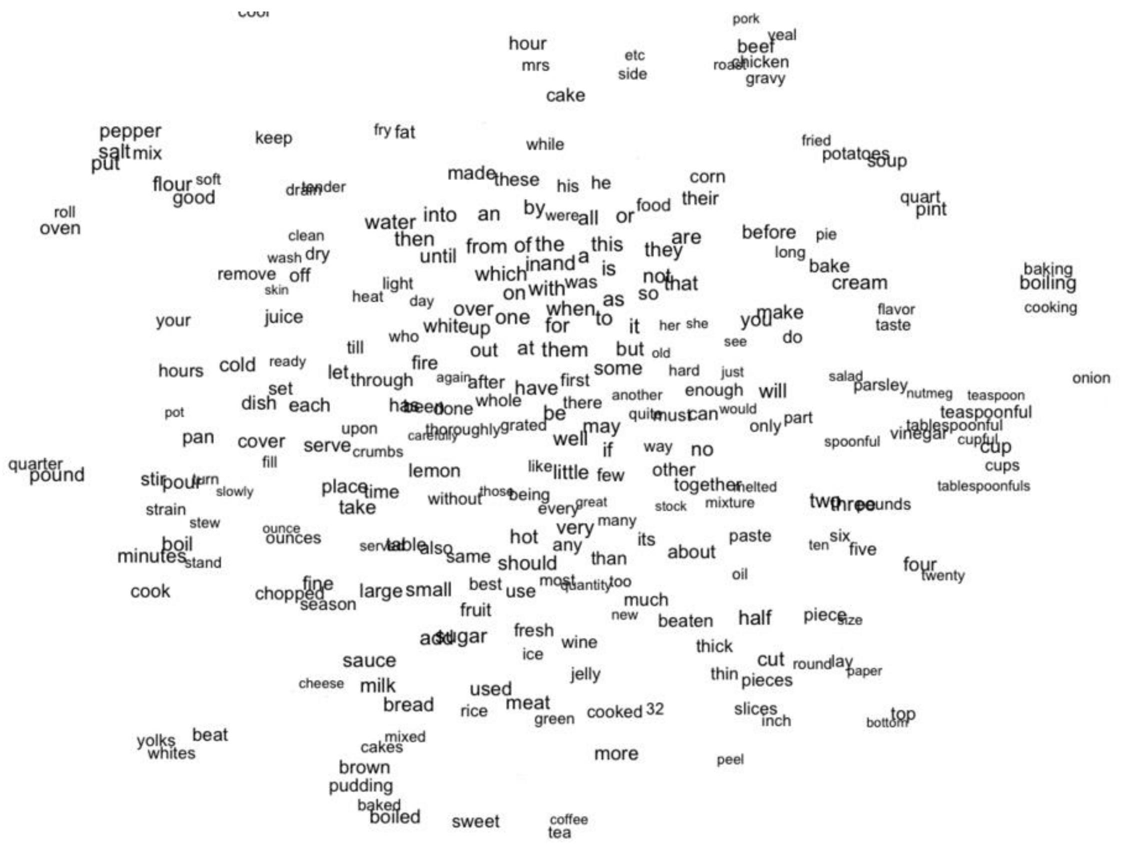
B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online Learning of Social Representations," *KDD*, 2014

Random Walk Applications

- Economics: Random walk hypothesis
- Genetics: Genetic drift
- Physics: Brownian motion
- Polymer Physics: Idea chain
- Computer Science: Estimate web size
- Image Segmentation
- …

Word2Vec

- Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In Advances in neural information processing systems, pp. 3111-3119. 2013.

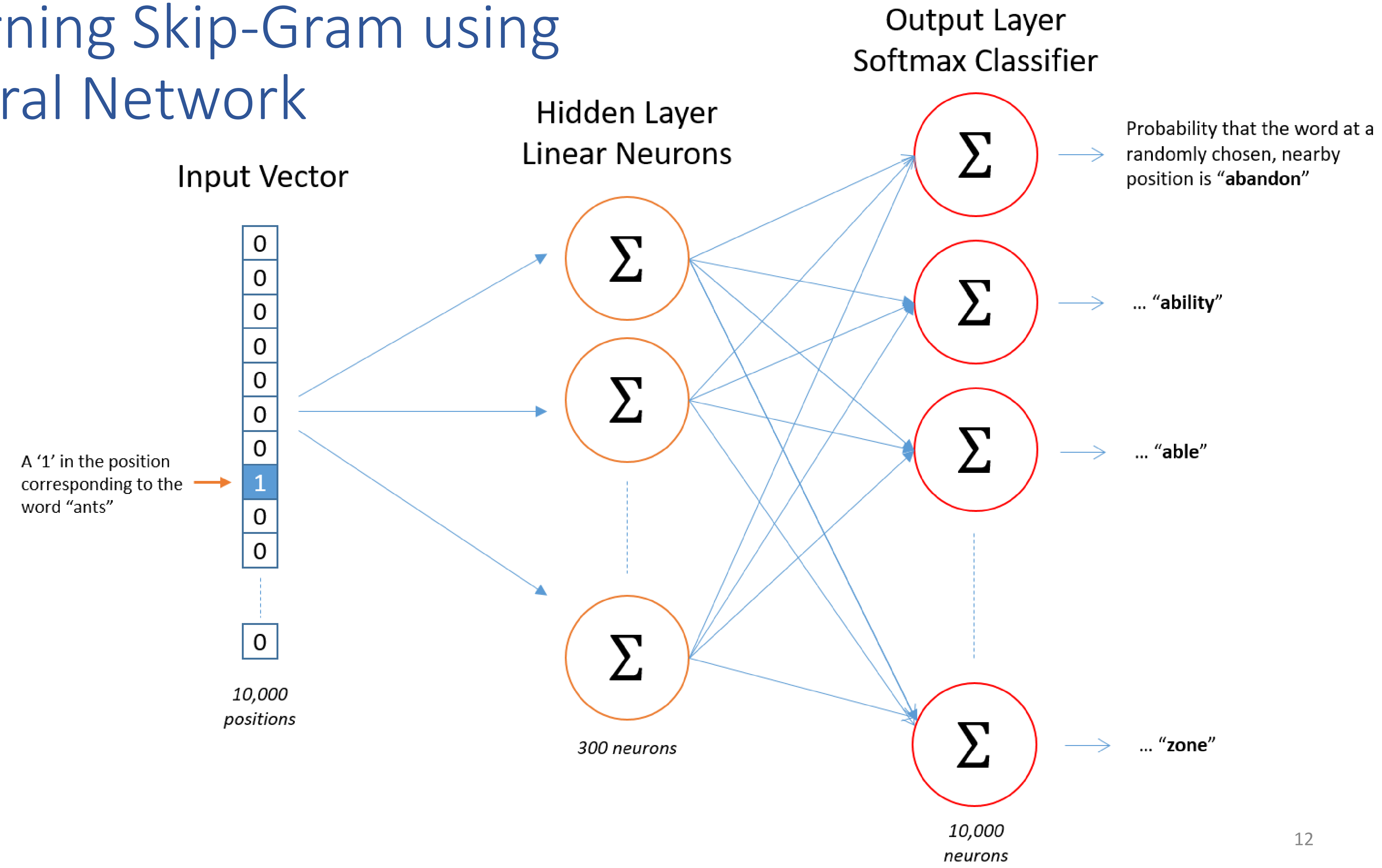


https://towardsdatascience.com/mapping-word-embeddings-with-word2vec-99a799dc9695

Skip-Gram Model

Source Text	Training Samples
The quick brown fox jumps over the lazy dog. ⟹	(the, quick) (the, brown)
The **quick** brown fox jumps over the lazy dog. ⟹	(quick, the) (quick, brown) (quick, fox)
The quick **brown** fox jumps over the lazy dog. ⟹	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown **fox** jumps over the lazy dog. ⟹	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

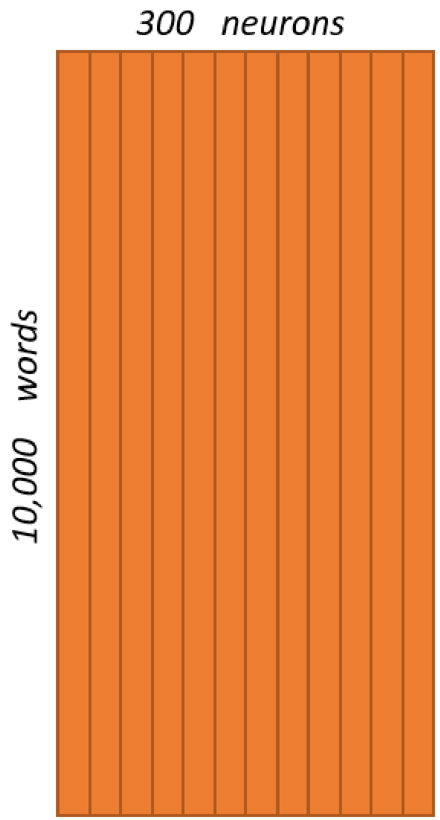
Learning Skip-Gram using Neural Network

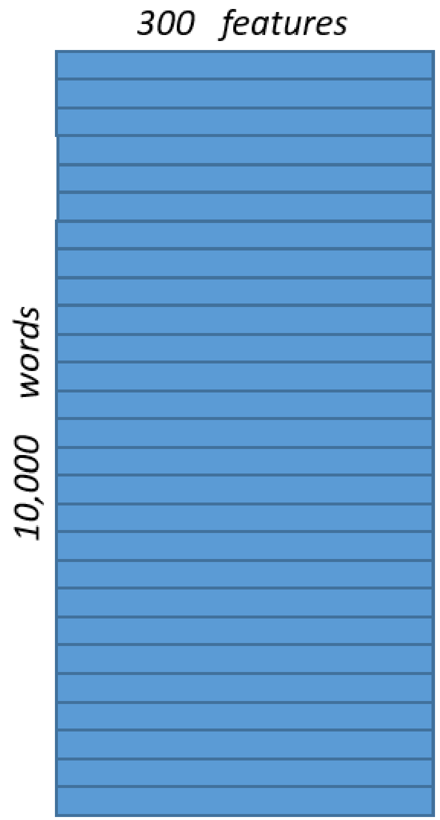


Using Weight of Hidden Neuron as Embedding Vectors

Hidden Layer
Weight Matrix

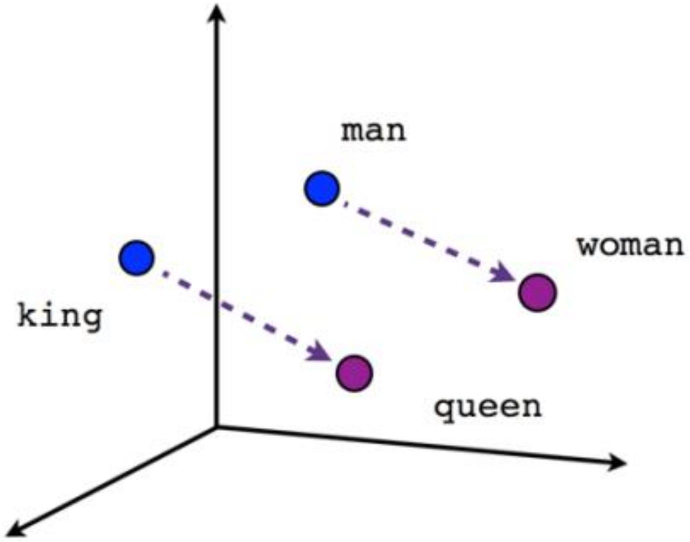
Word Vector
Lookup Table!



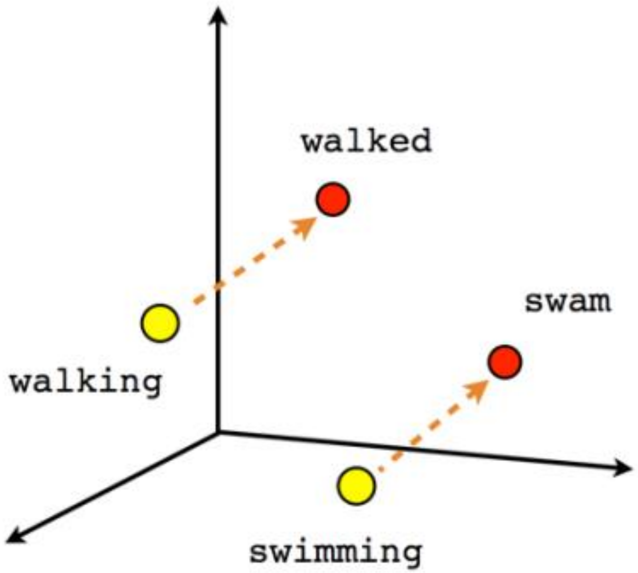


$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = \begin{bmatrix} 10 & 12 & 19 \end{bmatrix}$$

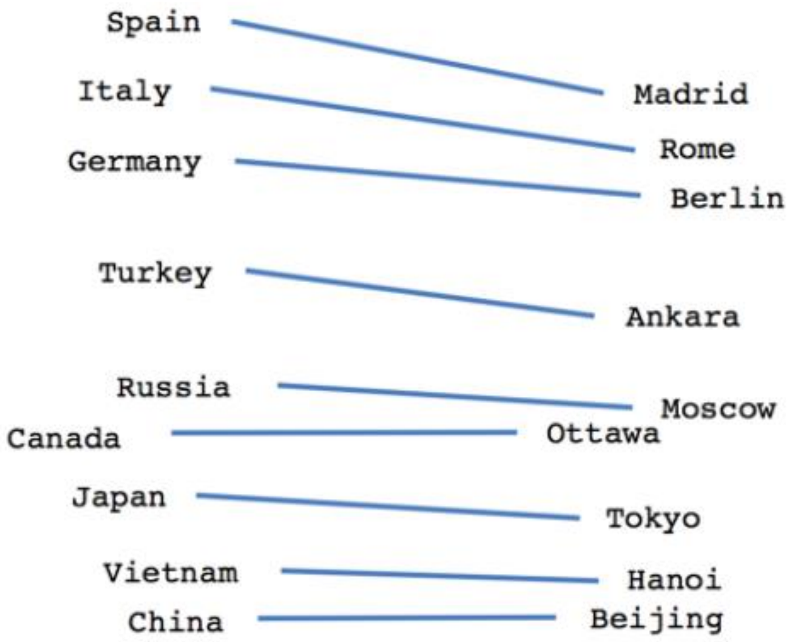
Evaluate Word2Vec



Male-Female



Verb tense



Country-Capital

Vector Addition & Subtraction

- vec(“Russia”) + vec(“river”) $\approx$ vec(“Volga River”)
- vec(“Germany”) + vec(“capital”) $\approx$ vec(“Berlin”)
- vec(“King”) - vec(“man”) + vec(“woman”) $\approx$ vec(“Queen”)

Datasets for Evaluating DeepWalk

- Blogs, Flicker, YouTube

Name	BlogCatalog	Flickr	YouTube
$\|V\|$	10,312	80,513	1,138,499
$\|E\|$	333,983	5,899,882	2,990,443
$\|\mathcal{Y}\|$	39	195	47
Labels	Interests	Groups	Groups

- Metric
  - Micro-F1
  - Macro-F1

Baseline Methods

- Spectral Clustering
  - Use *d*-smallest eigenvectors of normalized graph Laplacian of *G*
  - *Assume that graph cuts are useful for classification*
- Modularity
  - Select top-*d* eigenvectors of modular graph partitions of *G*
  - Assume that modular graph partitions are useful for classification
- Edge Cluster
  - Use k-means to cluster the adjacency matrix of G
- wvRN:
  - Weighted-vote Relational Neighbor
- Majority
  - The most frequent label

Classification Results in BlogCatalog

	% Labeled Nodes	10%	20%	30%	40%	50%	60%	70%	80%	90%
	DeepWalk	**36.00**	**38.20**	**39.60**	**40.30**	**41.00**	**41.30**	41.50	41.50	42.00
	SpectralClustering	31.06	34.95	37.27	38.93	39.97	40.99	**41.66**	**42.42**	**42.62**
	EdgeCluster	27.94	30.76	31.85	32.99	34.12	35.00	34.63	35.99	36.29
Micro-F1(%)	Modularity	27.35	30.74	31.77	32.97	34.09	36.13	36.08	37.23	38.18
	wvRN	19.51	24.34	25.62	28.82	30.37	31.81	32.19	33.33	34.28
	Majority	16.51	16.66	16.61	16.70	16.91	16.99	16.92	16.49	17.26
	DeepWalk	**21.30**	**23.80**	25.30	26.30	27.30	27.60	27.90	28.20	28.90
	SpectralClustering	19.14	23.57	**25.97**	**27.46**	**28.31**	**29.46**	**30.13**	**31.38**	**31.78**
	EdgeCluster	16.16	19.16	20.48	22.00	23.00	23.64	23.82	24.61	24.92
Macro-F1(%)	Modularity	17.36	20.00	20.80	21.85	22.65	23.41	23.89	24.20	24.97
	wvRN	6.25	10.13	11.64	14.24	15.86	17.18	17.98	18.86	19.57
	Majority	2.52	2.55	2.52	2.58	2.58	2.63	2.61	2.48	2.62

Classification Results in FLICKER

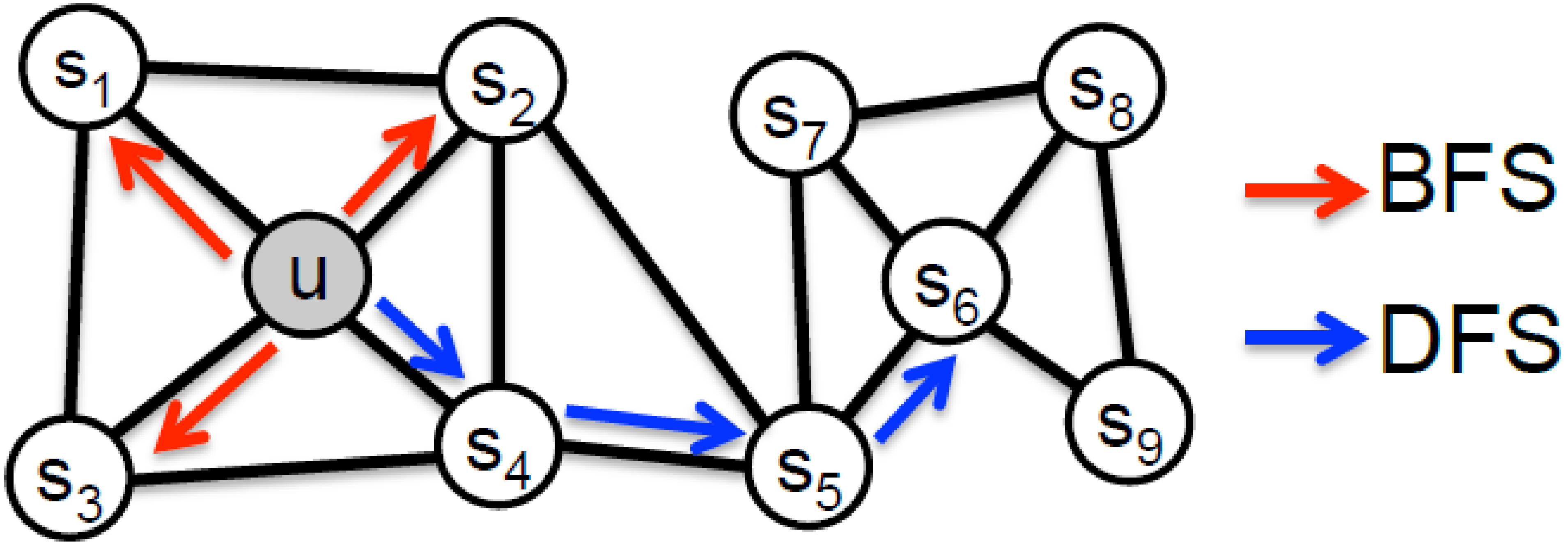
	% Labeled Nodes	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Micro-F1(%)	DeepWalk	**32.4**	**34.6**	**35.9**	**36.7**	**37.2**	**37.7**	**38.1**	**38.3**	**38.5**	**38.7**
	SpectralClustering	27.43	30.11	31.63	32.69	33.31	33.95	34.46	34.81	35.14	35.41
	EdgeCluster	25.75	28.53	29.14	30.31	30.85	31.53	31.75	31.76	32.19	32.84
	Modularity	22.75	25.29	27.3	27.6	28.05	29.33	29.43	28.89	29.17	29.2
	wvRN	17.7	14.43	15.72	20.97	19.83	19.42	19.22	21.25	22.51	22.73
	Majority	16.34	16.31	16.34	16.46	16.65	16.44	16.38	16.62	16.67	16.71
Macro-F1(%)	DeepWalk	**14.0**	17.3	**19.6**	**21.1**	**22.1**	**22.9**	**23.6**	**24.1**	**24.6**	**25.0**
	SpectralClustering	13.84	**17.49**	19.44	20.75	21.60	22.36	23.01	23.36	23.82	24.05
	EdgeCluster	10.52	14.10	15.91	16.72	18.01	18.54	19.54	20.18	20.78	20.85
	Modularity	10.21	13.37	15.24	15.11	16.14	16.64	17.02	17.1	17.14	17.12
	wvRN	1.53	2.46	2.91	3.47	4.95	5.56	5.82	6.59	8.00	7.26
	Majority	0.45	0.44	0.45	0.46	0.47	0.44	0.45	0.47	0.47	0.47

Classification Results in YouTube

	% Labeled Nodes	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Micro-F1(%)	DeepWalk	**37.95**	**39.28**	**40.08**	**40.78**	**41.32**	**41.72**	**42.12**	**42.48**	**42.78**	**43.05**
	SpectralClustering	—	—	—	—	—	—	—	—	—	—
	EdgeCluster	23.90	31.68	35.53	36.76	37.81	38.63	38.94	39.46	39.92	40.07
	Modularity	—	—	—	—	—	—	—	—	—	—
	wvRN	26.79	29.18	33.1	32.88	35.76	37.38	38.21	37.75	38.68	39.42
	Majority	24.90	24.84	25.25	25.23	25.22	25.33	25.31	25.34	25.38	25.38
Macro-F1(%)	DeepWalk	**29.22**	**31.83**	**33.06**	**33.90**	**34.35**	**34.66**	**34.96**	**35.22**	**35.42**	**35.67**
	SpectralClustering	—	—	—	—	—	—	—	—	—	—
	EdgeCluster	19.48	25.01	28.15	29.17	29.82	30.65	30.75	31.23	31.45	31.54
	Modularity	—	—	—	—	—	—	—	—	—	—
	wvRN	13.15	15.78	19.66	20.9	23.31	25.43	27.08	26.48	28.33	28.89
	Majority	6.12	5.86	6.21	6.1	6.07	6.19	6.17	6.16	6.18	6.19

Node2vec (2016)

- Homophily (communities) vs. Structure Equivalence (node roles)
- Add flexibility by exploring local neighborhoods
- Propose a biased random walk



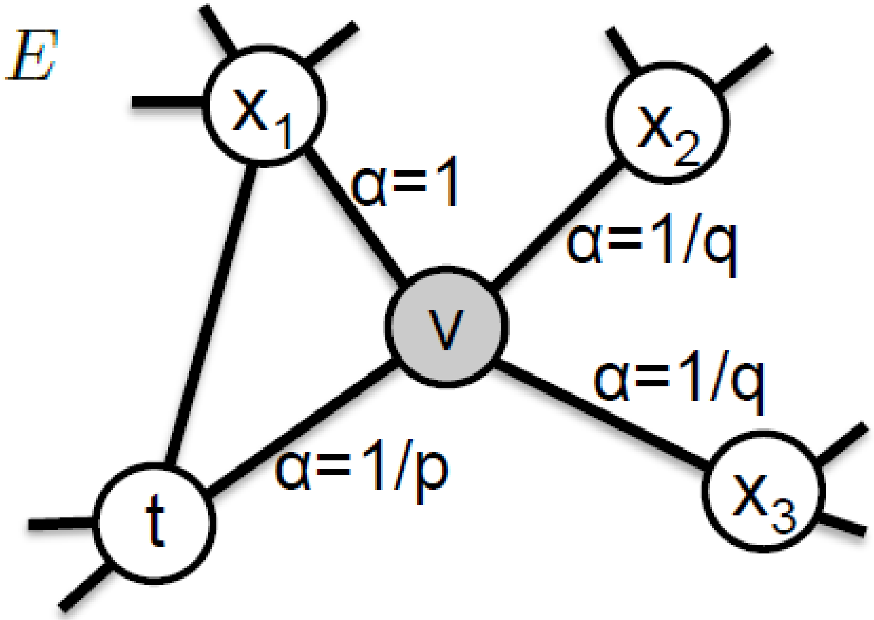
A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," *KDD*, 2016

Random walk with Bias α

- 3 directions: (1) return to previous node, (2) BFS, (3) DFS

$$P(c_i = x \mid c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & \text{if } (v, x) \in E \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$



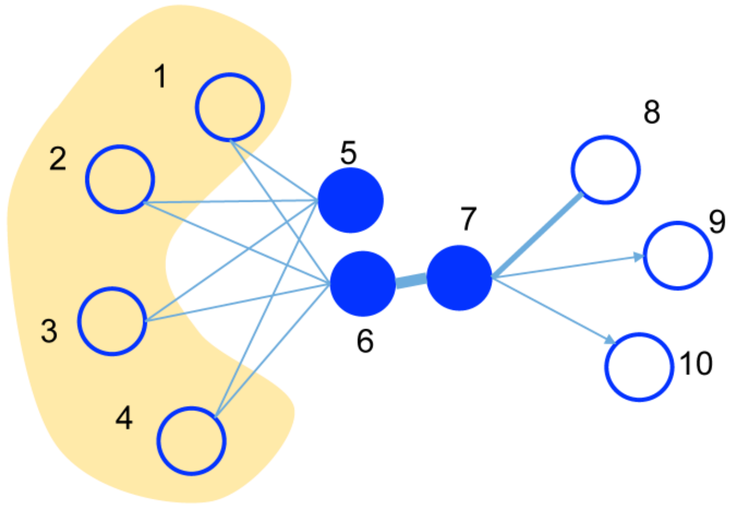
Experimental Results

	BlogCatalog	Protein-Protein Interactions (PPI)	Wikipedia
Vertices	10,312	3,890	4,777
Edges	333,983	76,584	184,812
Groups (Labels)	39	50	40

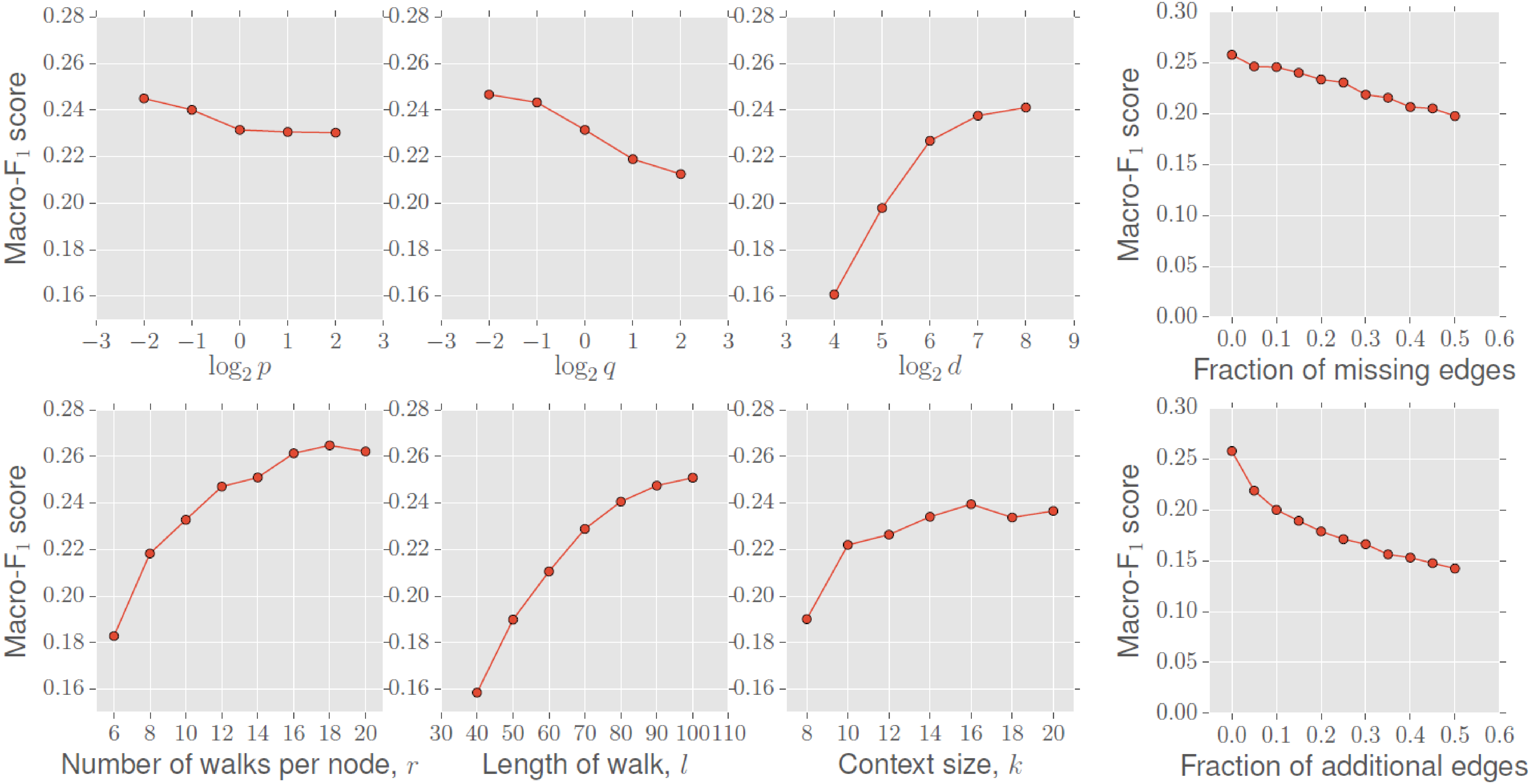
Algorithm	**Dataset**		
	BlogCatalog	PPI	Wikipedia
Spectral Clustering	0.0405	0.0681	0.0395
DeepWalk	0.2110	0.1768	0.1274
LINE	0.0784	0.1447	0.1164
node2vec	**0.2581**	**0.1791**	**0.1552**
node2vec settings (p,q)	0.25, 0.25	4, 1	4, 0.5
Gain of *node2vec* [%]	**22.3**	**1.3**	**21.8**

LINE: Large-scale Information Network Embedding

- J. Tang et al., "LINE: Large-scale Information Network Embedding," *WWW*, 2015
- Learn *d*-dimensional feature representations in two separate phases.
- In the first phase, it learns d=2 dimensions by BFS-style over neighbors.
- In the second phase, it learns the next d=2 dimensions by sampling nodes at a 2-hop distance from the source nodes.
  - Vertex 6 and 7 should be embedded closely as they are connected via a strong tie.
  - Vertex 5 and 6 should also be placed closely as they share similar neighbors.

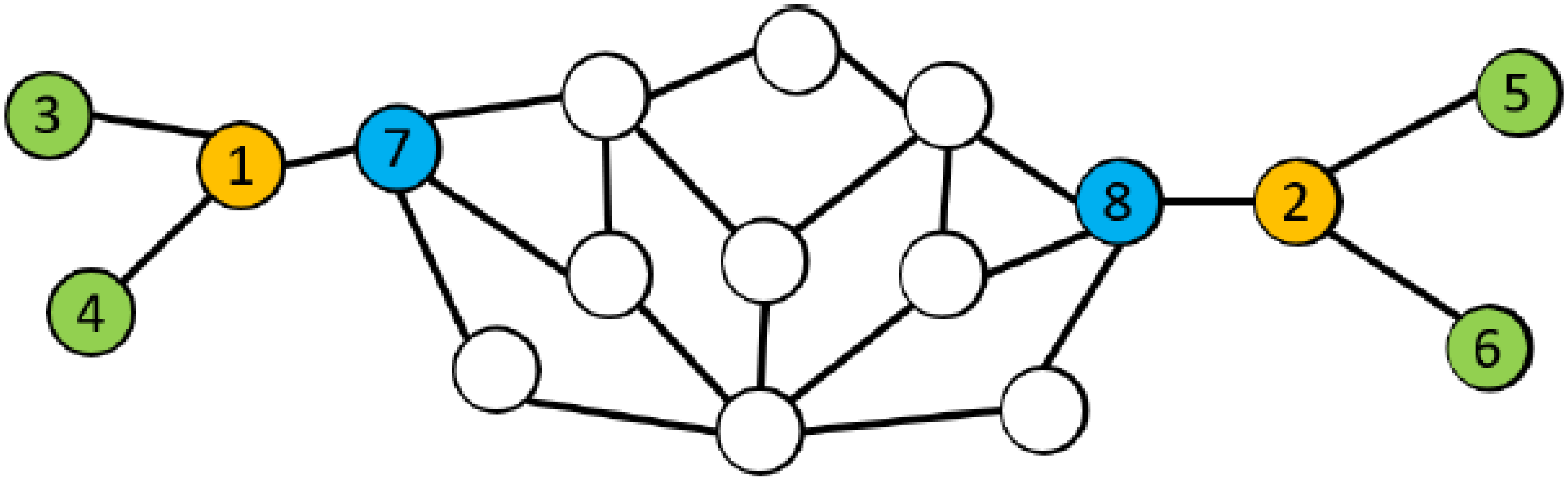


Parameters Sensitivity of node2vec



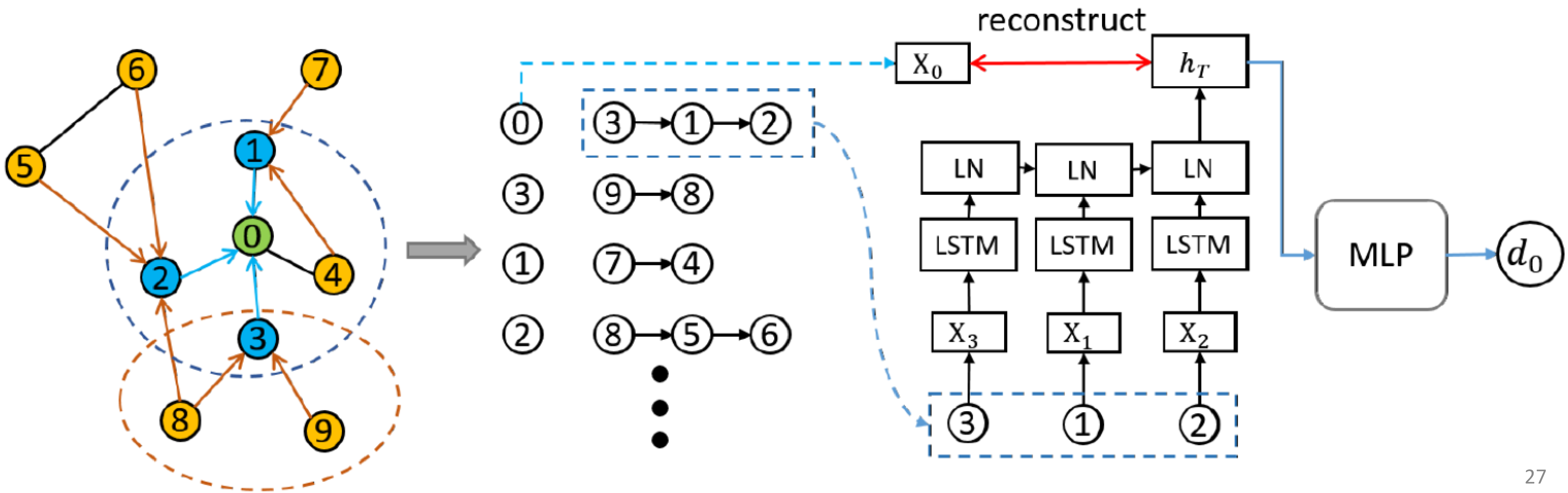
Deep Recursive Network Embedding with Regular Equivalence (2018)

- K. Tu, R. Cui, X. Wang, P. S. Yu, and W. Zhu, "Deep Recursive Network Embedding with Regular Equivalence," *KDD*, 2018



DRNE Brief Summary

- Sample and sort neighboring nodes by their degrees
- Encode nodes using layer-normalized LSTM



Who is the Boss? Identifying Key Roles in Telecom Fraud Network via Centrality-guided Deep Random Walk

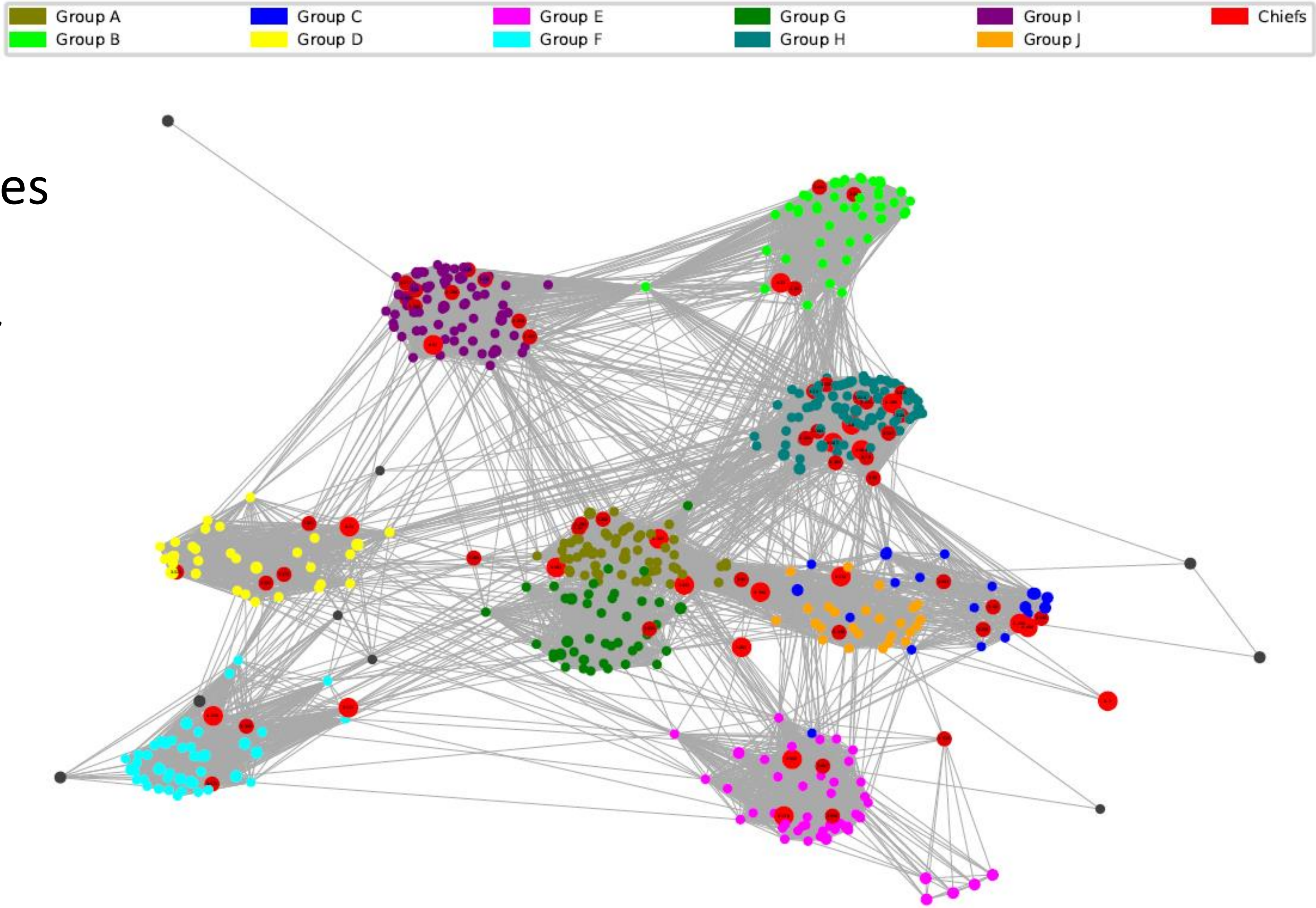
- Summitted to Social Networks (under review)
- Co-work with Criminal Investigation Bureau (CIB) in Taiwan


【形象大使】
ETtoday新聞雲

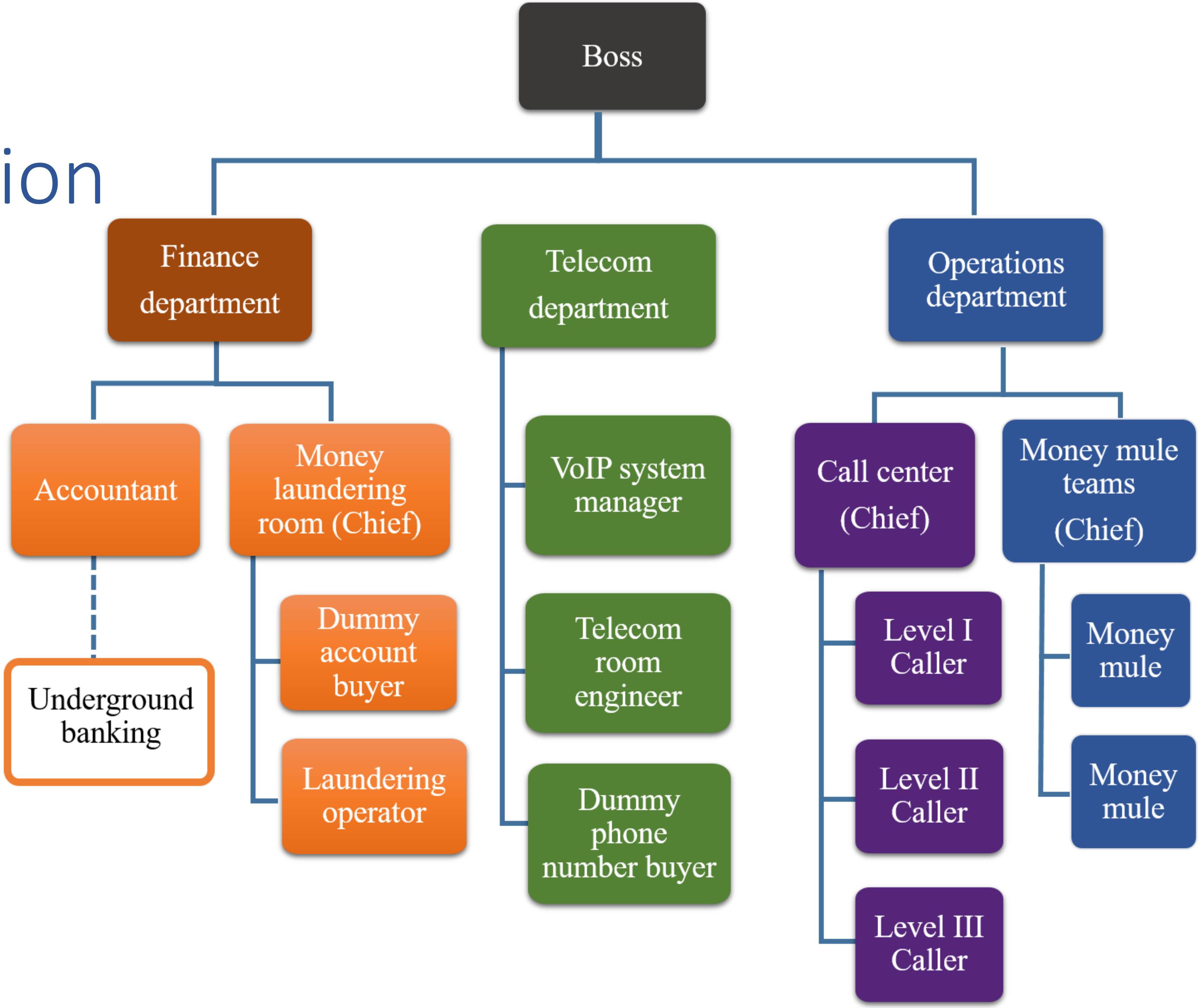
International Telecom Fraud

562 Fraudsters in 10 Groups

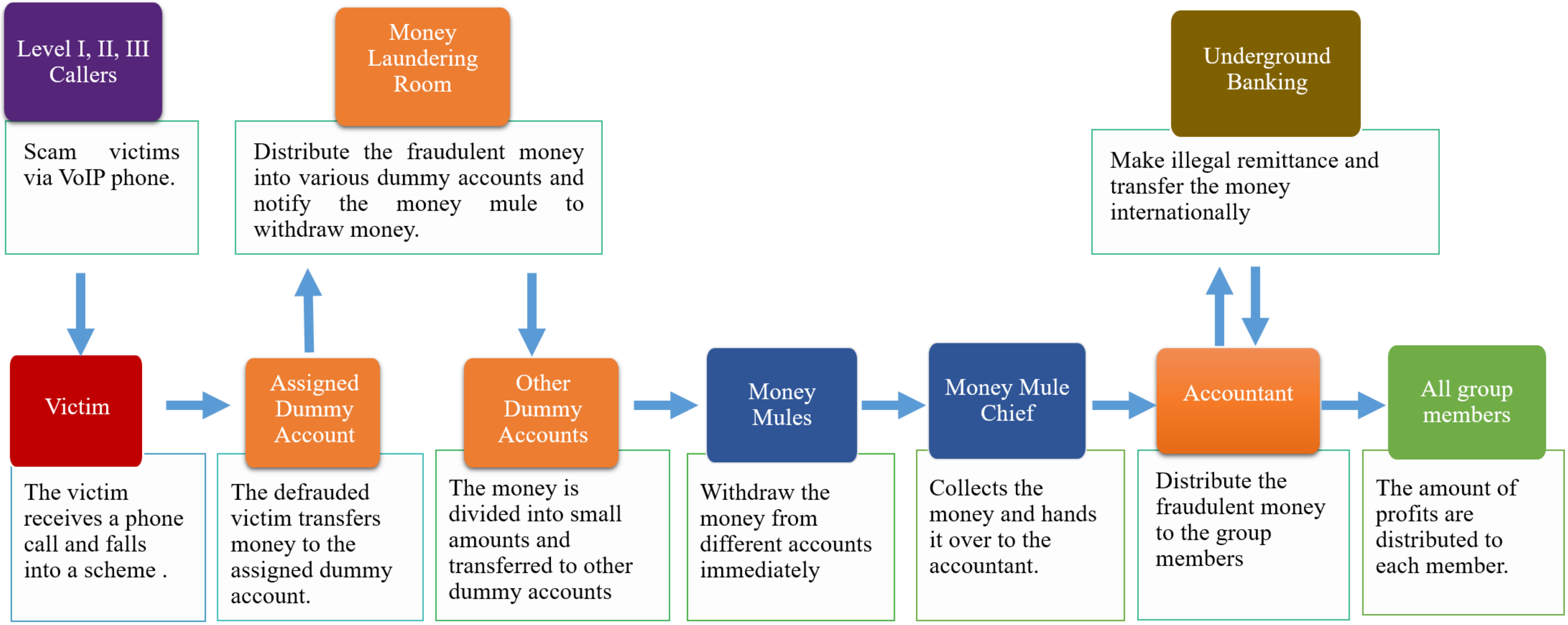
- Spread out in 17 cities of 4 countries
- *Linked via Co-offending records and flights*



Fraud Organization

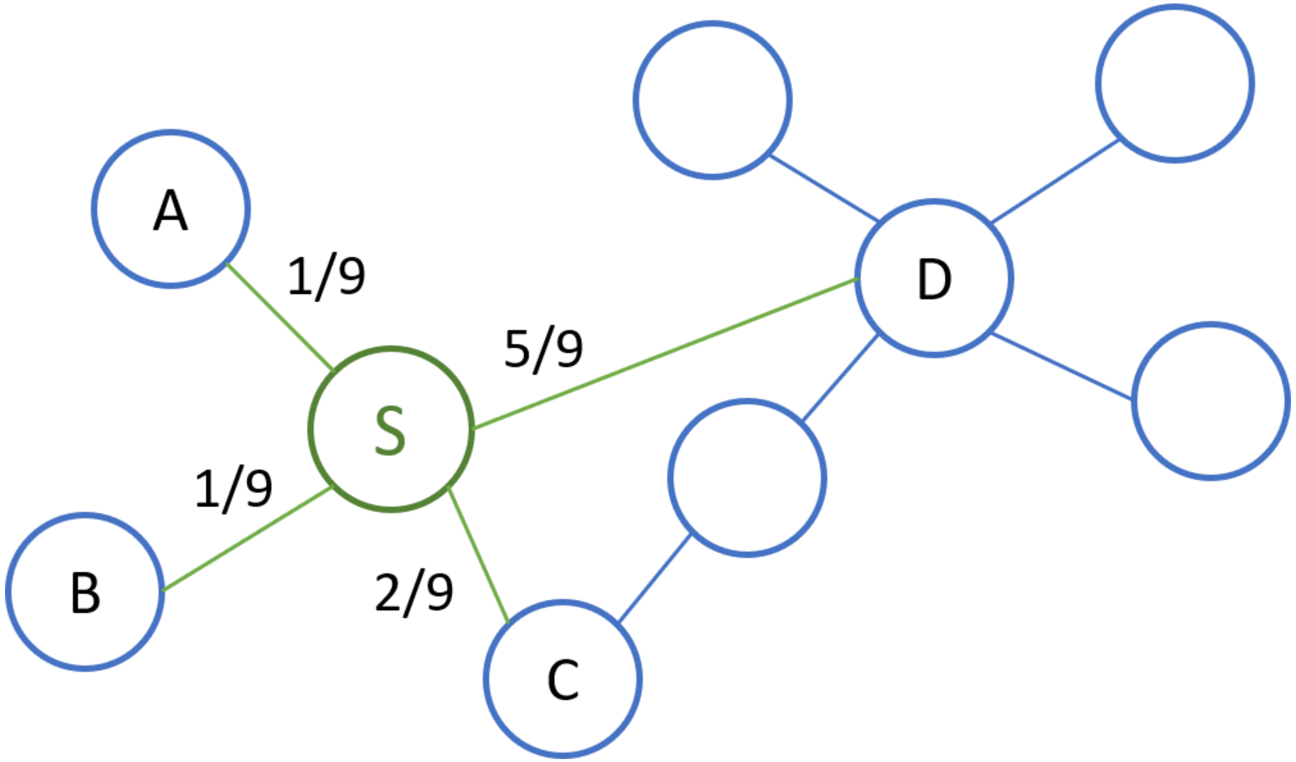


Telecom Fraud Flow



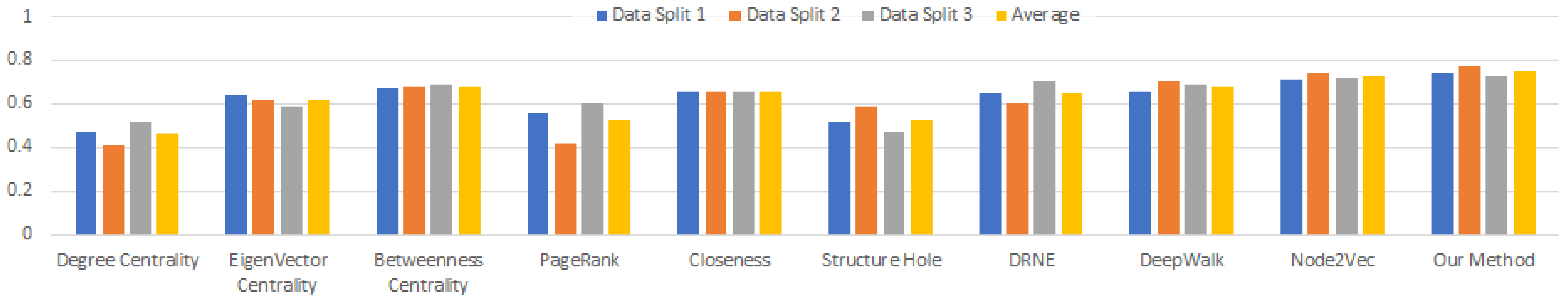
Centrality-guided Random Walk

- The neighbors of node S are nodes A, B, C, and D, which have degree centralities of 1, 1, 2, and 5



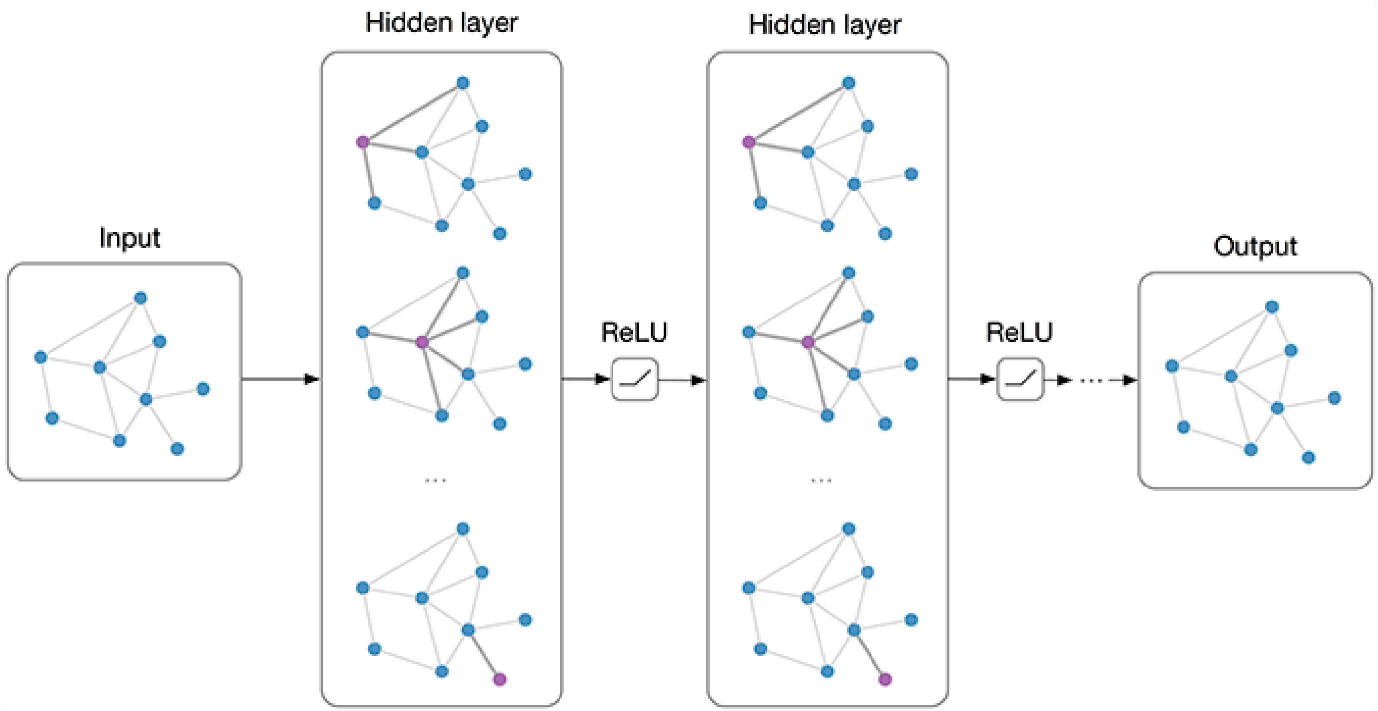
Experimental Results

Test Data	Degree Centrality	EigenVector Centrality	Betweenness Centrality	PageRank	Closeness Centrality	Structure Hole	DRNE	DeepWalk	Node2Vec	Our Method
Split 1	0.47	0.64	0.67	0.56	0.66	0.52	0.65	0.66	0.71	0.74
Split 2	0.41	0.62	0.68	0.42	0.66	0.59	0.60	0.69	0.74	0.77
Split 3	0.52	0.59	0.69	0.61	0.66	0.47	0.70	0.60	0.71	0.73
Average	0.47	0.62	0.68	0.53	0.66	0.53	0.65	0.68	0.72	0.75



GRAPH CONVOLUTIONAL NETWORKS (GCN)

- Thomas Kipf, 2016 (https://tkipf.github.io/graph-convolutional-networks/)
- Kipf & Welling (ICLR 2017), Semi-Supervised Classification with Graph Convolutional Networks
- Defferrard et al. (NIPS 2016), Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering



GCN Formula

- Given a graph *G=(V,E)*
- Xi for every node i; summarized in a N×D feature matrix $X \in \mathbb{R}^{N \times D}$
  - N: number of nodes
  - D: dimension of input features
- *A* is the adjacency matrix *A* of G
- Output $Z \in \mathbb{R}^{N \times F}$, F is the dimension of output features

$$H^{(l+1)} = \sigma\left(AH^{(l)}W^{(l)}\right)$$

Addressing Limitations

- Normalizing the adjacency matrix A via graph Laplacian
  - $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, D is the degree matrix

- Add self-loop to use its own feature as input
  - $\tilde{A} = A + I$

$$H^{(l+1)} = \sigma\left(D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

Graph Convolution for Hashtag Recommendation

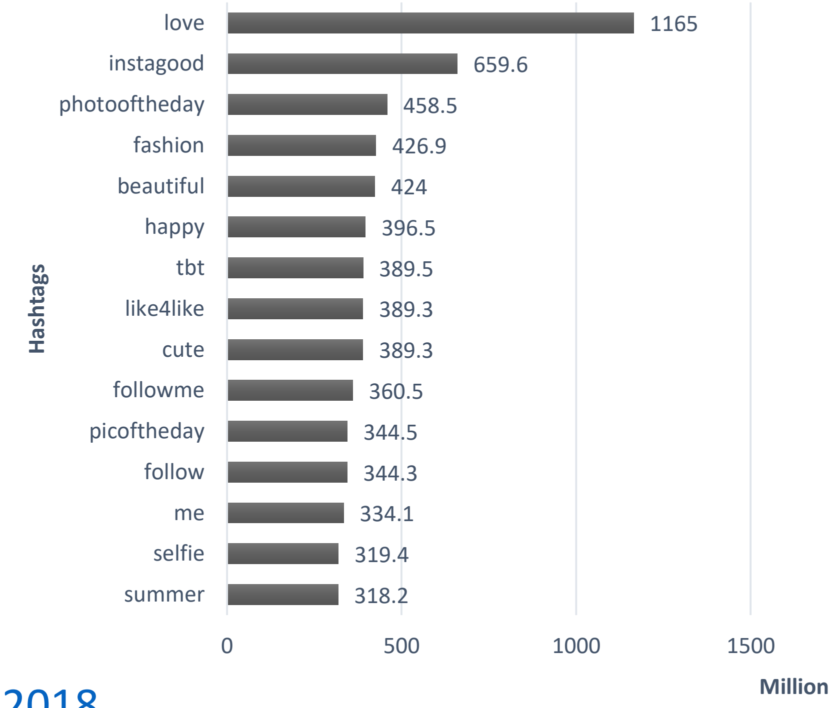
2019.10.28

Student: Yu-Chi Chen(Judy)

Advisors: Prof. Ming-Syan Chen, Kuan-Ting Lai

Image Hashtag Recommendation

- Hashtag => a word or phrase preceded by the symbol # that categorizes the accompanying text
- Created by Twitter, now supported by all social networks
- Instagram hashtag statistics (2017):



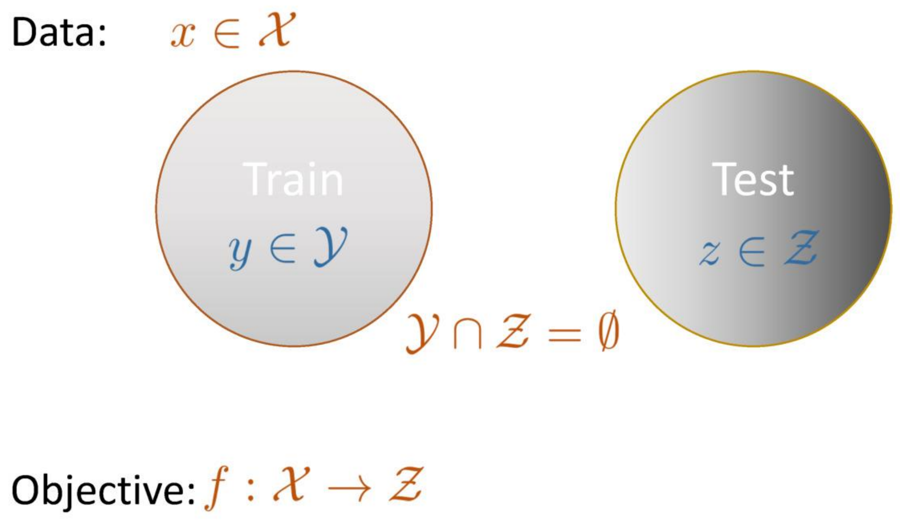
[Latest stats: izea.com/2018/06/07/top-instagram-hashtags-2018]()

Difficulties of Predicting Image Hashtag

- Abstraction: #love, #cute,...
- Abbreviation: #ootd, #ootn,…
- Emotion: #happy,…
- Obscurity: #motivation, #lol,…
- New-creation: #EvaChenPose,…
- No-relevance: #tbt, #nofilter, #vscocam
- Location: #NYC, #London

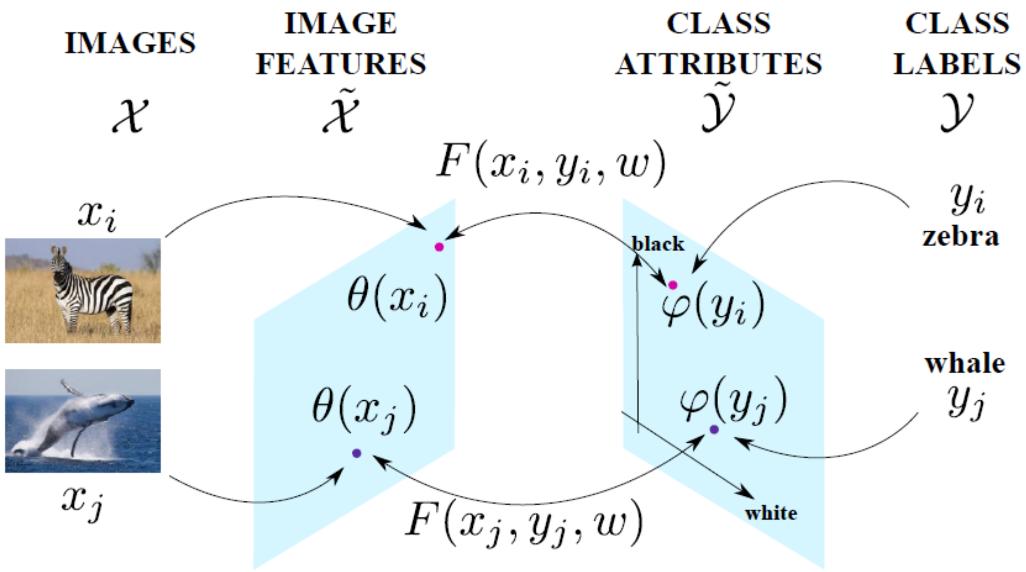
Zero-Shot Learning

- Identify object that you've never seen before
- More formal definition:
  - Classify test classes *Z* with zero labeled data (Zero-shot!)



Zero-Shot Formulation

- Describe objects by words
  - Use attributes (semantic features)

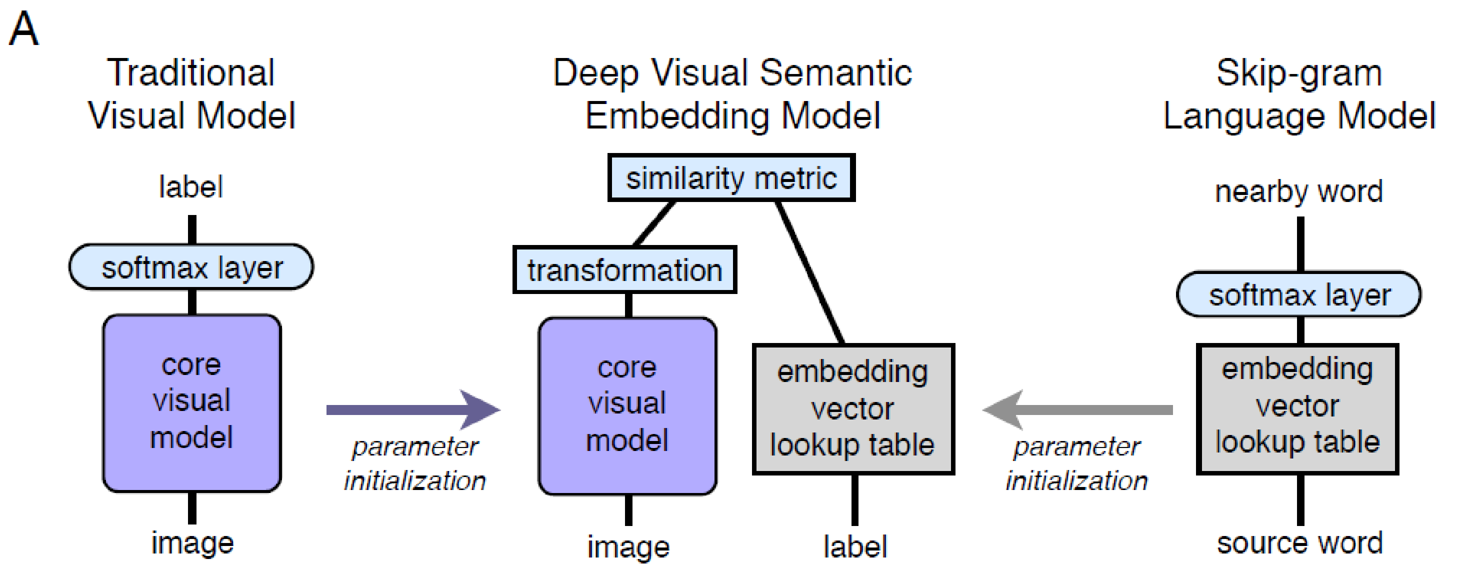


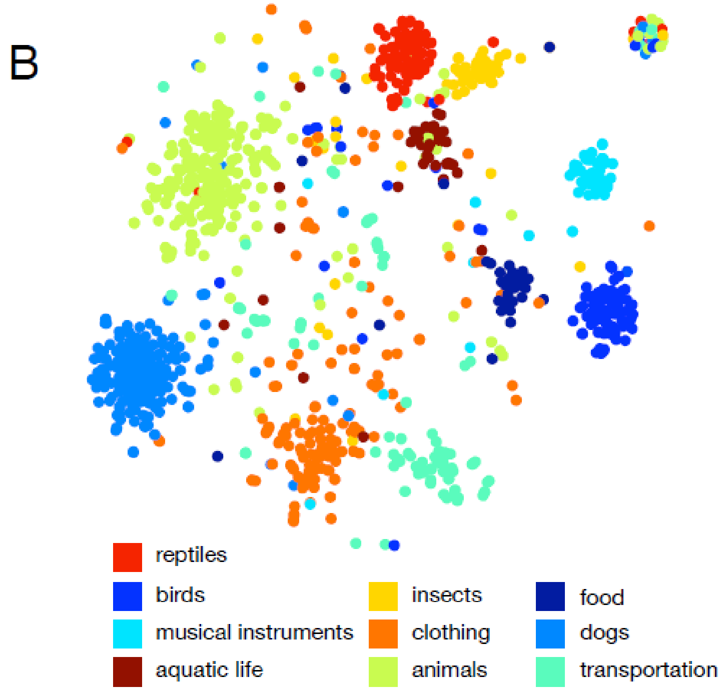
$$F(x, y; W) = \theta(x)^T W \phi(y)$$

DeViSE – Deep Visual Semantic Embedding

- Google, NIPS, 2013

Model	200 labels	1000 labels
DeViSE	31.8%	9.0%
Mensink et al. 2012 [12]	35.7%	1.9%
Rohrbach et al. 2011 [17]	34.8%	-

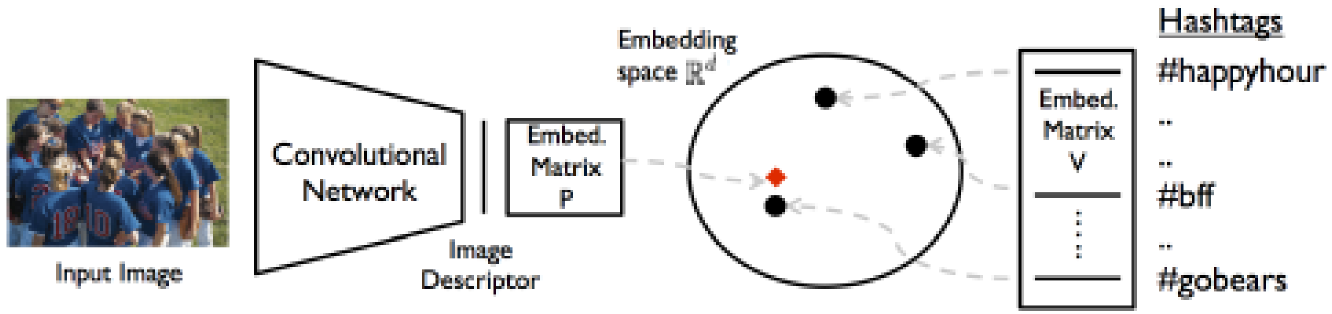




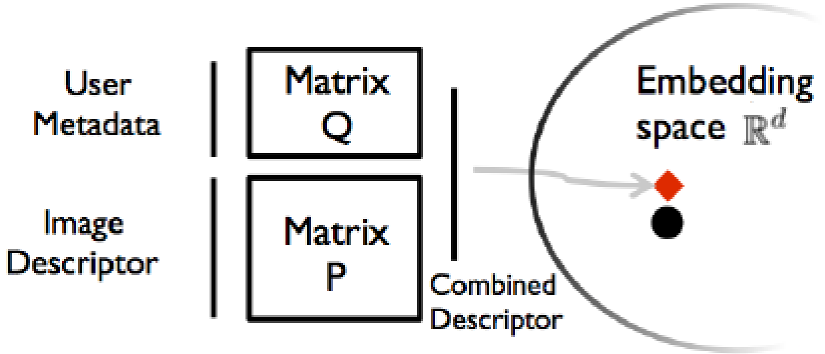
State-of-the-art: User Conditional Hashtag Prediction for Images

- E. Denton, J. Weston, M. Paluri, L. Bourdev, and R. Fergus, "User Conditional Hashtag Prediction for Images," *ACM SIGKDD*, 2015 (Facebook)
- Hashtag Embedding: $f(x, y) = \Phi_I(x)^\top \Phi_H(y)$
- Proposed 3 models:

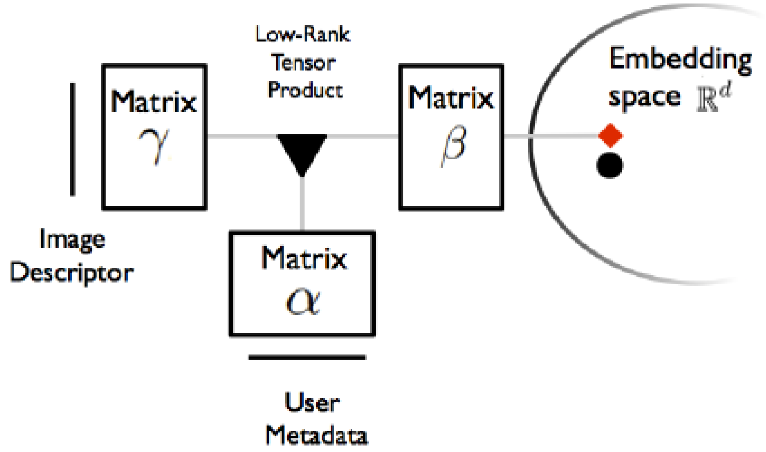
1. Bilinear Embedding Model



2. User-biased model



3. User-multiplicative model



Age	Females	Males
13-17	#mcm	#like
	#bestfriend	#lmp
	#love	#throwback
	#lovehim	#squad
	#mce	#wce
	#latepost	#throwback-
	#bestfriends	#thursday
	#boyfriend	#family
	#loveher	#workflow
	#loveyou	#selfie
		#wcm
43-47	#100happydays	#photoshop-
	#mcm	#express
	#love	#wcw
	#sisters	#goodtimes
	#cousins	#prouddad
	#lovehim	#throwback-
	#latergram	#thursday
	#loveher	#selfie
	#bff	#salute
	#youcampperfect	#blessed
		#zijasummit14
		#familyfirst

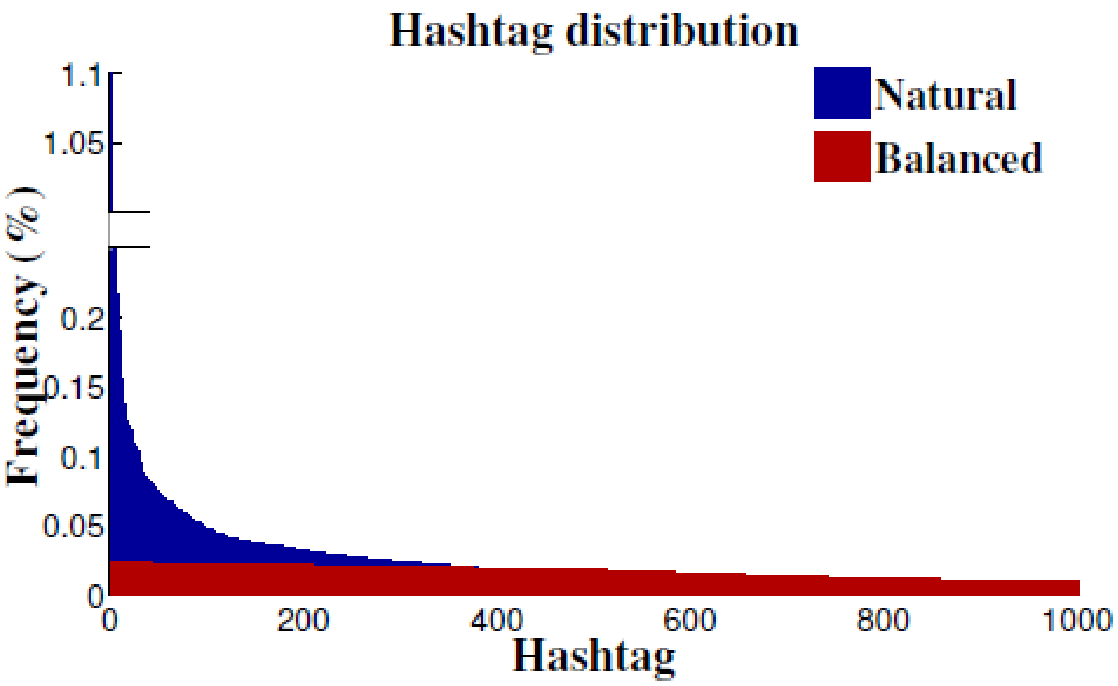
Gender	15-17 years old	43-47 years old
Female	#wcw	#100happydays
	#mcm	#blessed
	#bestfriend	#goodtimes
	#tb	#family
	#ss	#love
	#bestfriends	#photogrid
	#throwback	#latergram
	#latepost	#cousins
	#like	#sundayfunday
	#selfiesunday	#friends
Male	#wcw	#goodtimes
	#like	#blessed
	#throwback	#love
	#squad	#family
	#tb	#photoshop-
	#lmp	#express
	#mcm	#photogrid
	#ss	#sundayfunday
	#wce	#friends
	#selfiesunday	#zijasummit14
		#prouddad

Sydney	Toronto
#melbourne	#toronto
#sydney	#tbt
#australia	#canada
#spring	#vancouver
#beach	#fall
#grandfinal	#throwback
#sunshine	#blessed
#sun	#ilovethiscity
#nz	#vancity
#newzealand	#vscocam
#bali	#tb
#happy	#cntower
#nofilter	#goodmorning
#wellington	#hoco
#springbreak	#montreal
#bondi	#wcw
#afl	#tdot
#thailand	#lateupload
#stkilda	#downtown
#city	#beautiful

Meta data	Possible values
Age	13 – 114
Gender	Male, Female, Unknown
Home City	GPS coordinates
Country	United States, Canada, Great Britain, Australia, New Zealand

Facebook's Experiments

- 20 million images
- 4.6 million hashtags, average 2.7 tags per image
- Result



Method	d	K	P@1	R@10	A@10
Freq. baseline	-	-	3.04%	5.63%	9.45%
Bilinear	64	-	7.37%	11.71%	18.69%
Bilinear	128	-	7.37%	11.69%	18.44%
Bilinear	256	-	6.75%	10.84%	17.25%
Bilinear	512	-	6.50%	10.83%	17.17%
User-biased	64	-	9.02%	13.63%	21.88%
User-biased	128	-	9.00%	13.67%	21.83%
User-biased	256	-	8.48%	13.03%	20.96%
User-biased	512	-	7.98%	12.51%	20.05%
3-way mult.	64	50	8.95%	13.66%	21.82%
3-way mult.	64	100	9.03%	13.81%	22.04%
3-way mult.	64	200	8.96%	13.81%	22.05%
3-way mult.	64	300	9.00%	13.74%	21.96%
3-way mult.	64	400	8.96%	13.65%	21.82%

My Work

Introduction

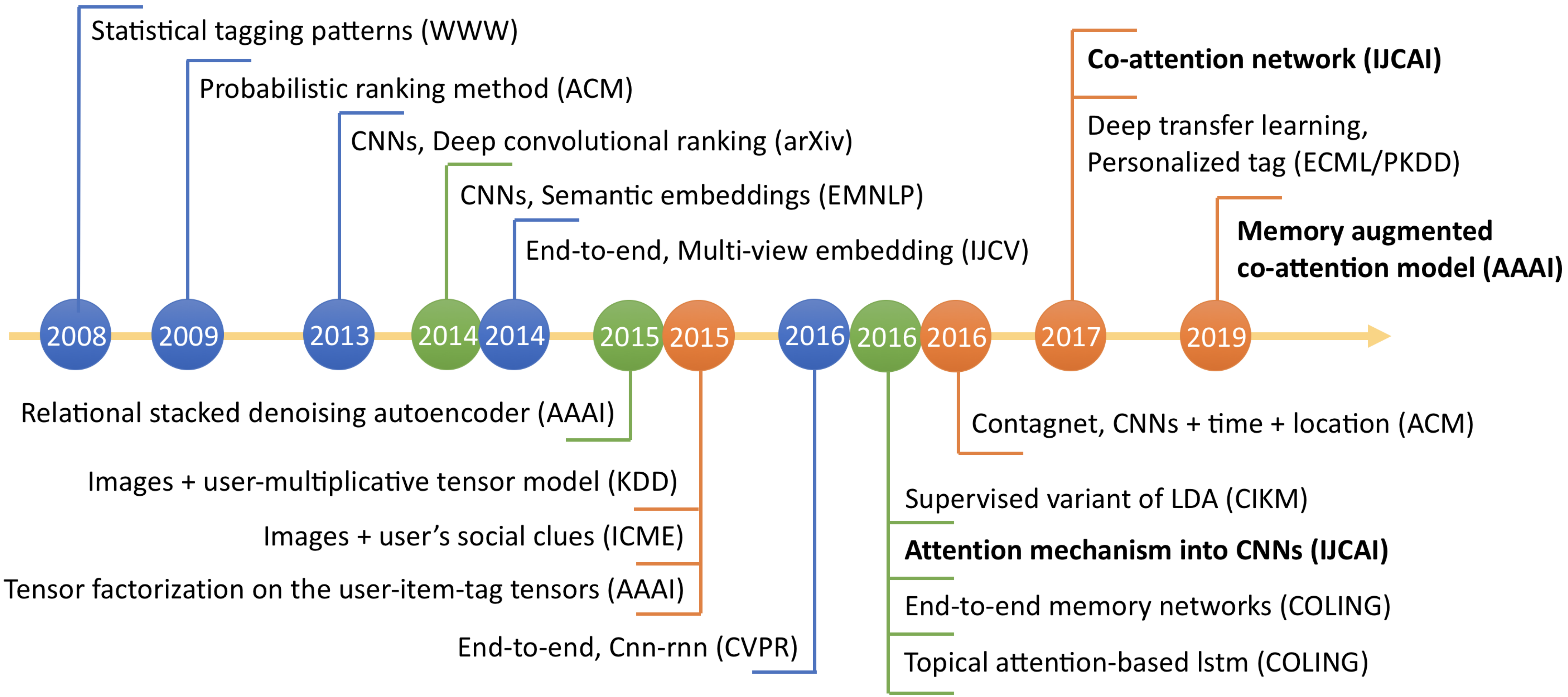
- **Goal**:
  - Given information of IG posts, including images and texts, the goal is to recommend corresponding hashtags.
- **Main contribution**:
  - Use multiple types of input and implement graph convolution network for hashtag recommendation.
- **Dataset**: MaCon
  - Every post has some attributes: post_id, words, hashtags, user_id, images.

#Posts	#Users	#Hashtags	Ave_p	Ave_h
624,520	7,497	3896	83.3	6.41

Average posts of a user.

Related Work Overview

Based on images
Based on text
Based on multimodal data



Related Work Overview

Based on images
Based on text
Based on multimodal data

Statistical tagging patterns (WWW)

Probabilistic ranking method (ACM)

2008 2009

CNNs, Deep convolutional ranking (arXiv)

Co-attention network (IJCAI)

End-to-end, Cnn-rnn (CVPR)

Topical attention-based lstm (COLING)

Statistical tagging patterns: Sigurbjörnsson, B., and Van Zwol, R. 2008. **Flickr tag recommendation based on collective knowledge.** In *WWW*, 327–336.

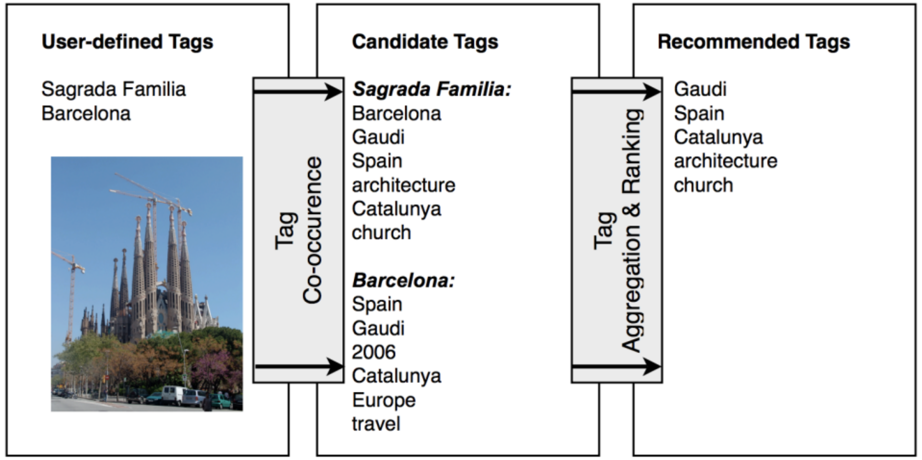


Figure 4: System overview of the tag recommendation process.

Related Work Overview

Based on images
Based on text
Based on multimodal data

Statistical tagging patterns (WWW)

Probabilistic ranking method (ACM)

CNNs, Deep convolutional ranking (arXiv)

Co-attention network (IJCAI)

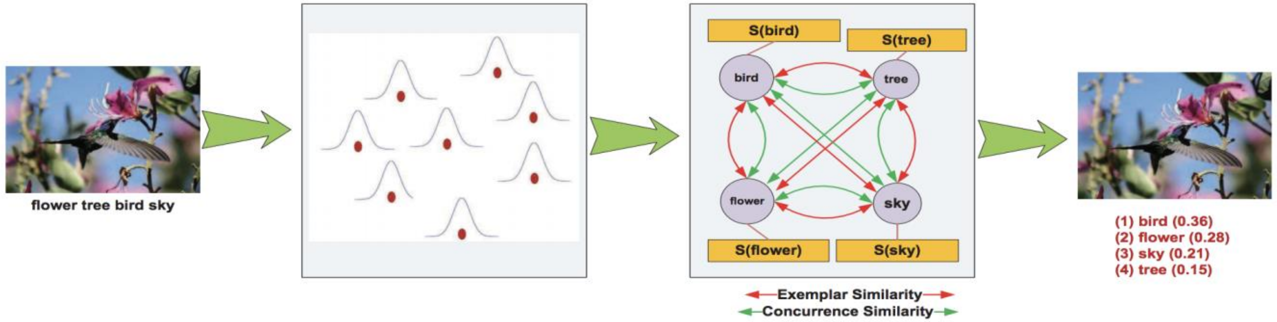
Deep transfer learning, Personalized tag (ECML/PKDD)

Memory augmented co-attention model (AAAI)

End-to-end, Multi-view embedding (IJCV)

2008 2009 2013

Probabilistic ranking method: Liu, D.; Hua, X.-S.; Yang, L.; Wang, M.; and Zhang, H.-J. 2009. **Tag ranking.** In *WWW*, 351–360. ACM.



Relational stacked denoising autoe

Images + user-multiplicative

Images + user'

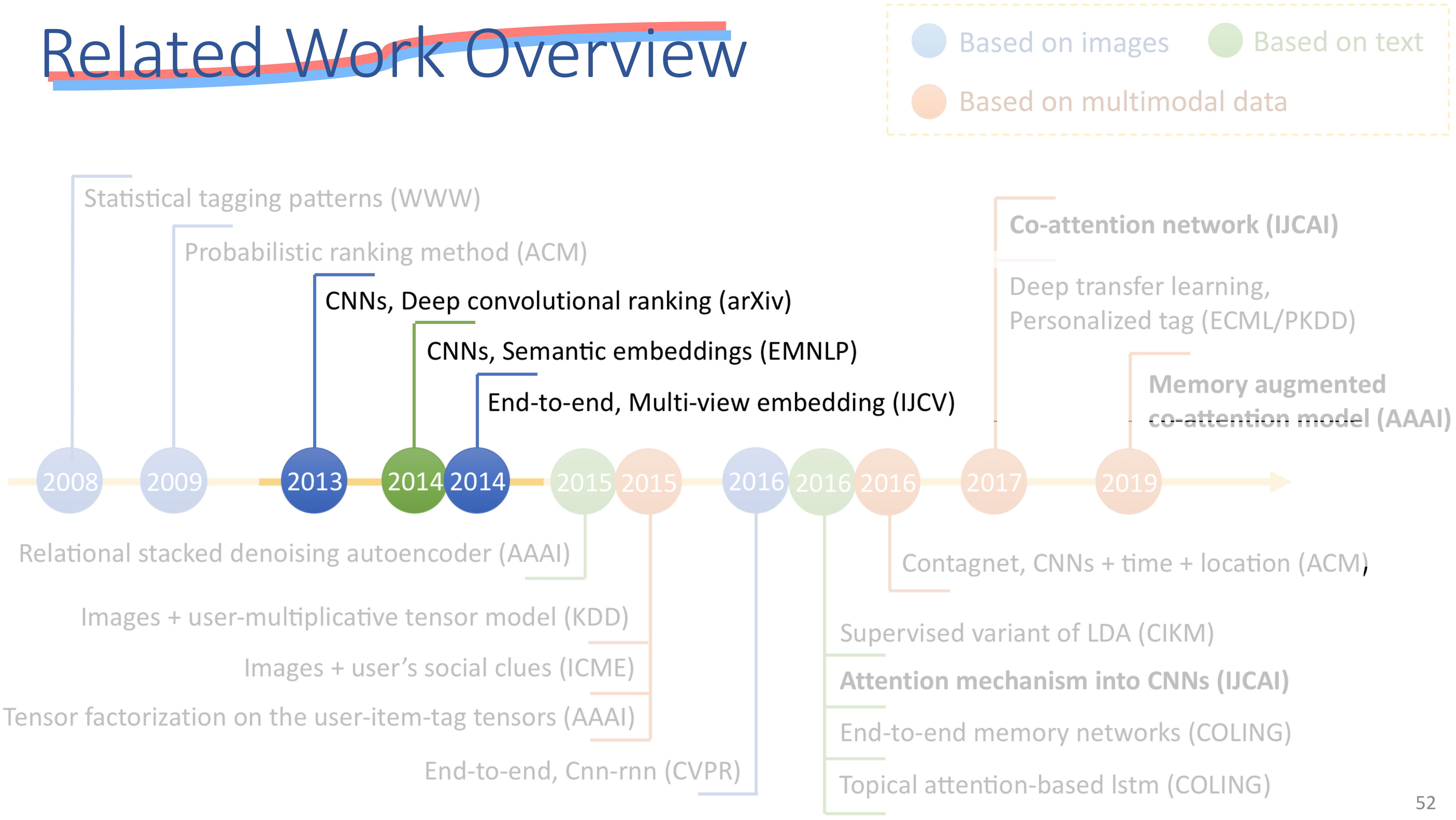
Tensor factorization on the user-item-tag tensors (AAAI)

End-to-end memory networks (COLING)

End-to-end, Cnn-rnn (CVPR)

Topical attention-based lstm (COLING)

Related Work Overview



Related Work Overview

Based on images
Based on text
Based on multimodal data

Statistical tagging patterns (WWW)

Probabilistic ranking method (ACM)

CNNs, Deep convolutional ranking

CNNs, Semantic embedd

End-to-end, Multi-view embedd

2008 2009 2013 2014 2014 2015 2015 2016 2016

Relational stacked denoising autoencoder (AAAI)

Images + user-multiplicative tensor model (KDD)

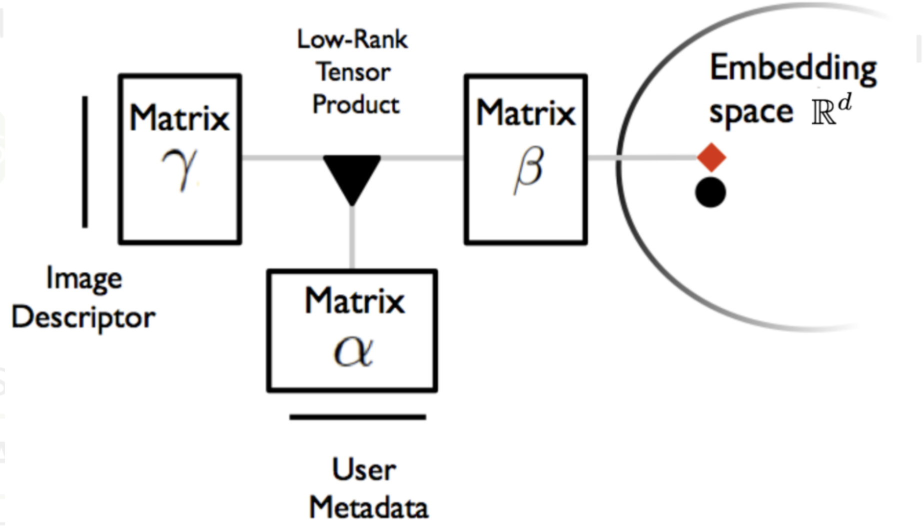
Images + user's social clues (ICME)

Tensor factorization on the user-item-tag tensors (AAAI)

End-to-end, Cnn-rnn (CVPR)

Topical attention-based lstm (COLING)

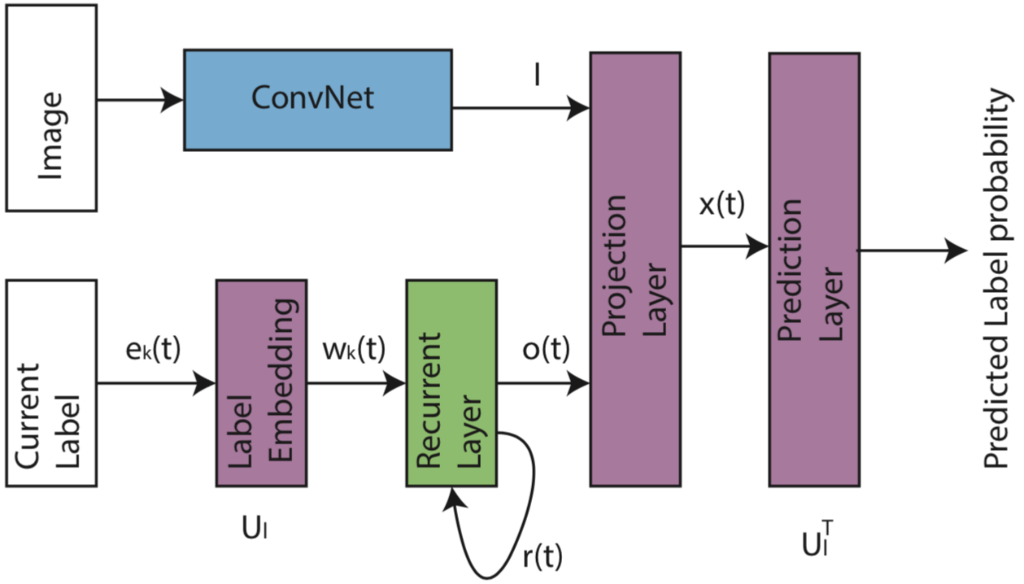
Images + user-multiplicative tensor model: Denton, E.; Weston, J.; Paluri, M.; Bourdev, L.; Fergus, R. 2015. **User conditional hashtag prediction for images.** In: Proceedings of the SIGKDD Conference on Knowledge Discovery and Data Mining.



Related Work Overview

Based on images
Based on text
Based on multimodal data

End-to-end model: Wang, J.; Yang, Y.; Mao, J.; Huang, Z.; Huang, C.; and Xu, W. 2016. **Cnn-rnn: A unified framework for multi-label image classification.** In *CVPR*, 2285–2294.



CNNs, Deep convolutional ranking (arXiv)
dings (EMNLP)
view embedding (IJCV)
Tensor factorization on the user-item-tag tensors (AAAI)

End-to-end, Cnn-rnn (CVPR)

2016 2016 2016 2017 2019

Co-attention network (IJCAI)

Deep transfer learning, Personalized tag (ECML/PKDD)

Memory augmented co-attention model (AAAI)

Contagnet, CNNs + time + location (ACM)

Supervised variant of LDA (CIKM)

Attention mechanism into CNNs (IJCAI)

End-to-end memory networks (COLING)

Topical attention-based lstm (COLING)

Related Work Overview

Based on images
Based on text
Based on multimodal data

Attention mechanism into CNNs: Gong, Y., and Zhang, Q. 2016. **Hashtag recommendation using attention-based convolutional neural network.** In *IJCAI*, 2782–2788.

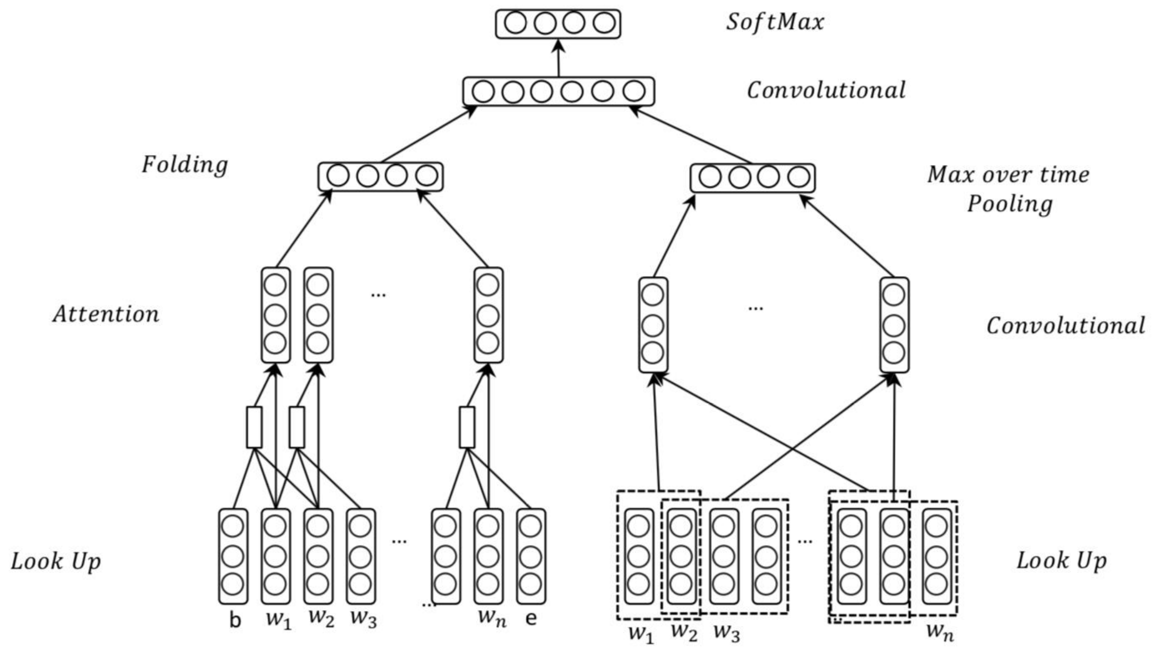
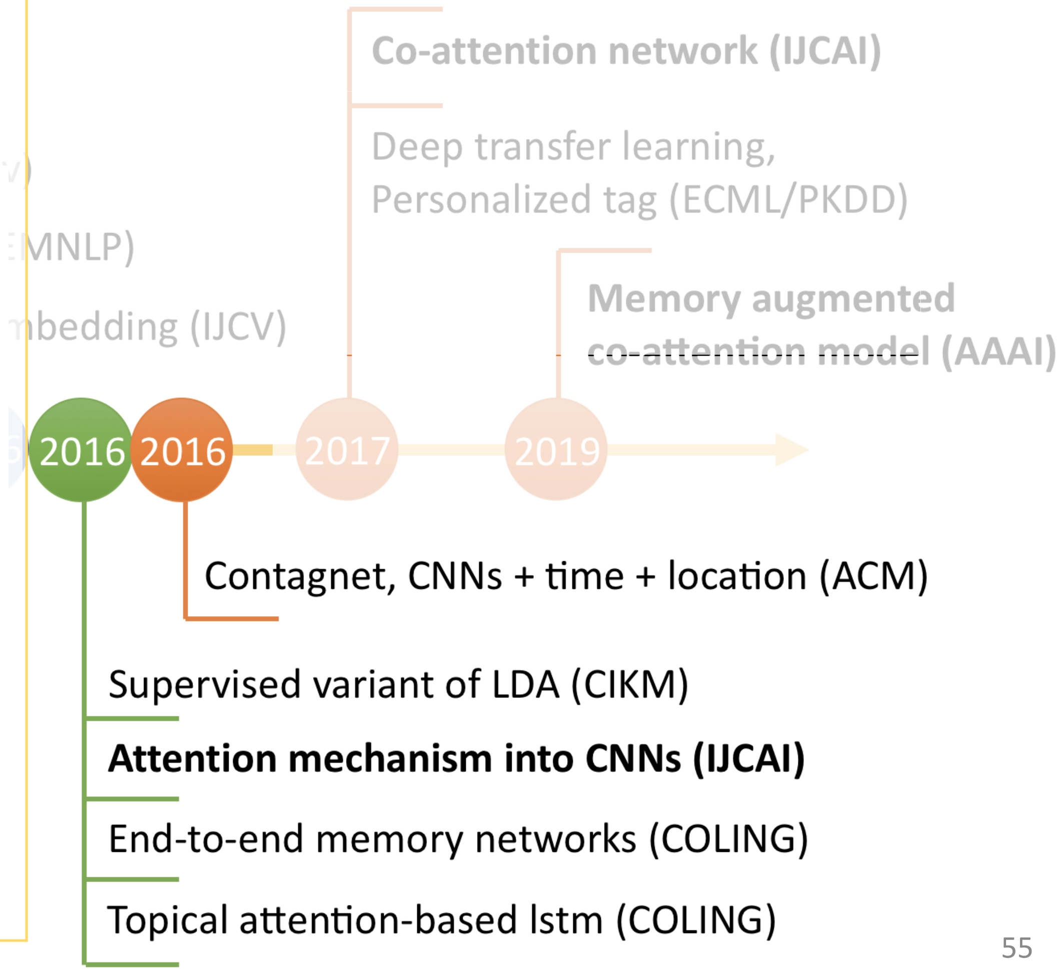
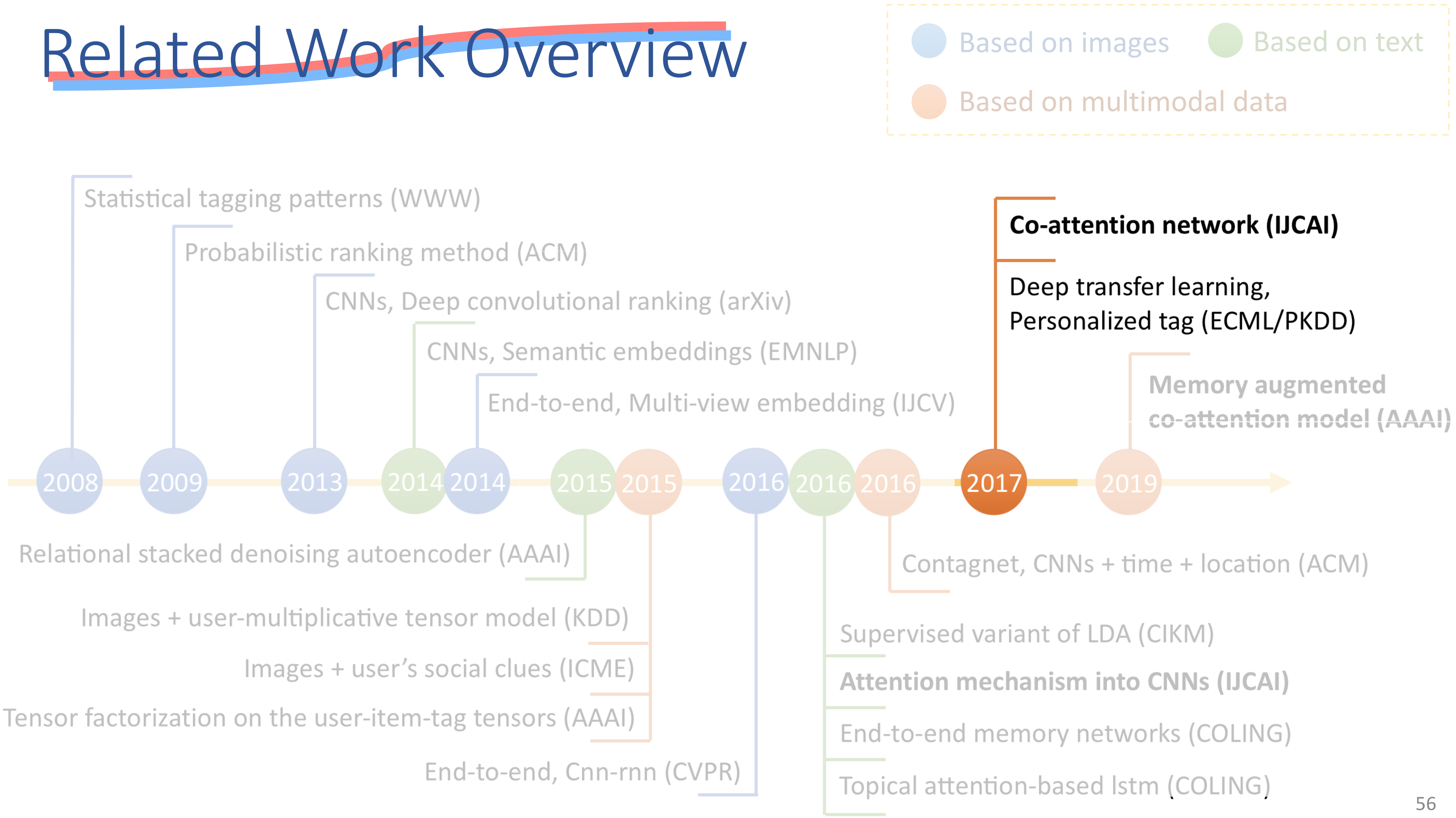


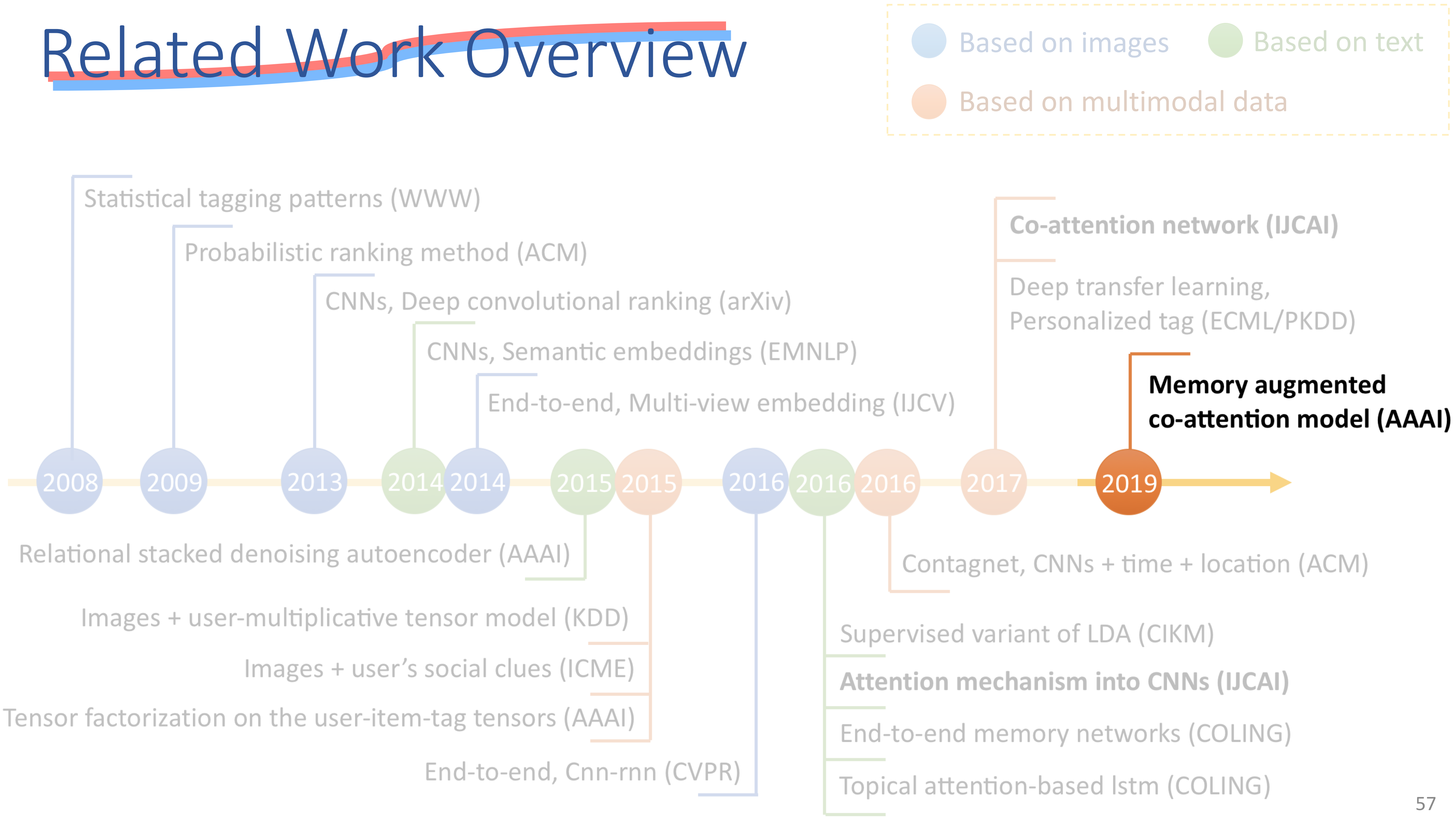
Figure 1: The architecture of the attention-based Convolutional Neural Network



Related Work Overview



Related Work Overview

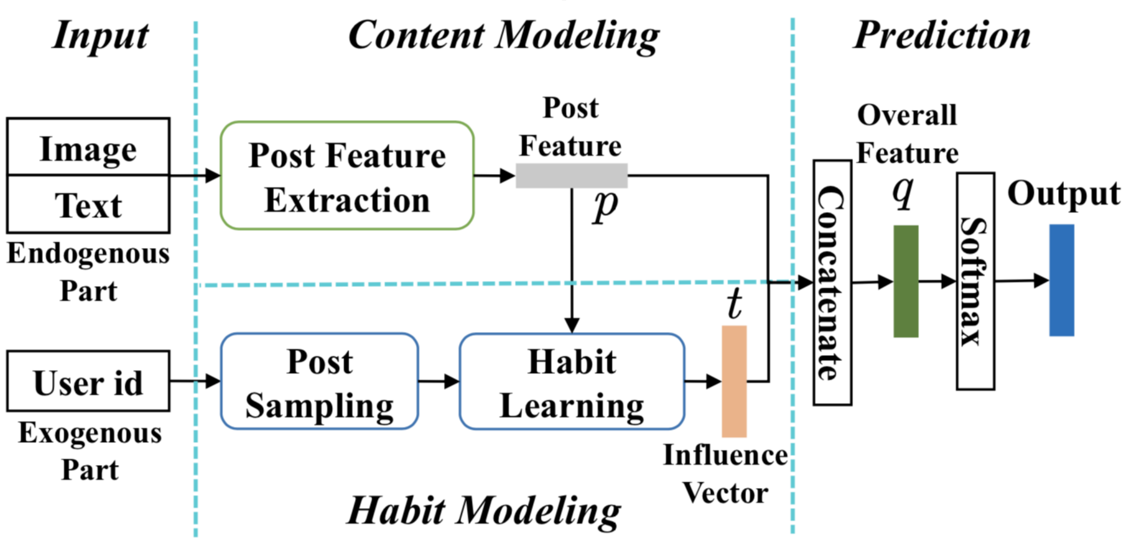


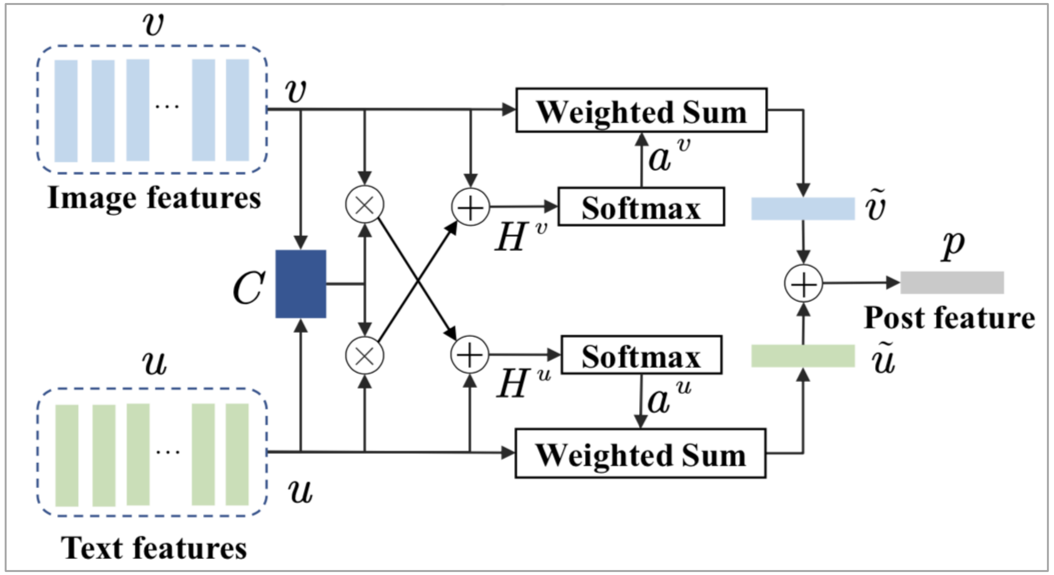
Related work

Hashtag Recommendation for Photo Sharing Services

Suwei Zhang[1], Yuan Yao[1], Feng Xu[1], Hanghang Tong[2], Xiaohui Yan[3], Jian Lu[1]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Arizona State University, USA [3]Poisson Lab, Huawei Technologies, China
zsw@smail.nju.edu.cn, {y.yao, xf, lj}@nju.edu.cn, hanghang.tong@asu.edu, yanxiaohui2@huawei.com

- 2019 AAAI. Memory Augmented CO-attentioN model (**MACON**)
- Multi-label classification problem





parallel co-attention mechanism

external memory unit

Dataset: MaCon

- Every post has some attributes: post_id, words, hashtags, user_id, images (40G).

#Posts	#Users	#Hashtags	Ave_p	Ave_h
624,520	7,497	3896	83.3	6.41

- Paper: (from 2019 AAAI)

Hashtag Recommendation for Photo Sharing Services

Suwei Zhang[1], Yuan Yao[1], Feng Xu[1], Hanghang Tong[2], Xiaohui Yan[3], Jian Lu[1]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Arizona State University, USA [3]Poisson Lab, Huawei Technologies, China
zsw@smail.nju.edu.cn, {y.yao, xf, lj}@nju.edu.cn, hanghang.tong@asu.edu, yanxiaohui2@huawei.com

Related work

Learning Context Graph for Person Search

Yichao Yan[1,2,3,4*] Qiang Zhang[1*] Bingbing Ni[1†]
Wendong Zhang[2] Minghao Xu[1] Xiaokang Yang[2]
[1]Shanghai Jiao Tong University, China
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China
[3] Tencent YouTu Lab, China [4] Inception Institute of Artificial Intelligence, UAE
{yanyichao, zhangqiang2016, nibingbing, diergent, xuminghao118, xkyang}@sjtu.edu.cn

- 2019 CVPR
- Person search (end-to-end human detection + multi-part feature learning)
- **Build a graph** to learn global similarity between two individuals considering context information

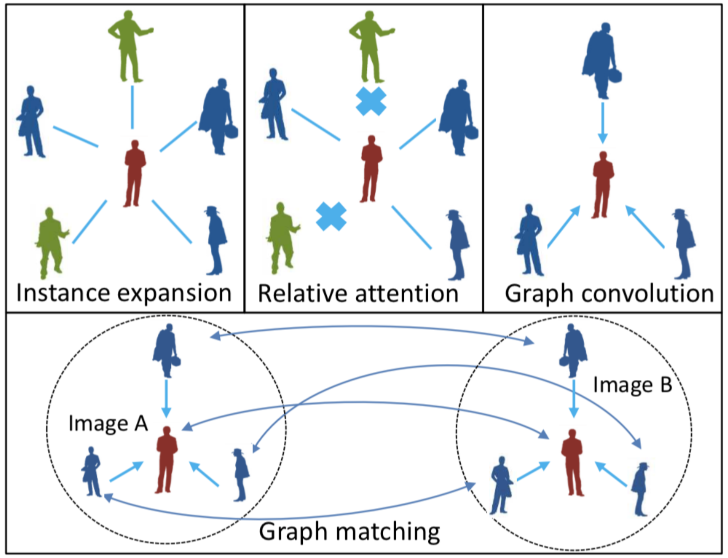


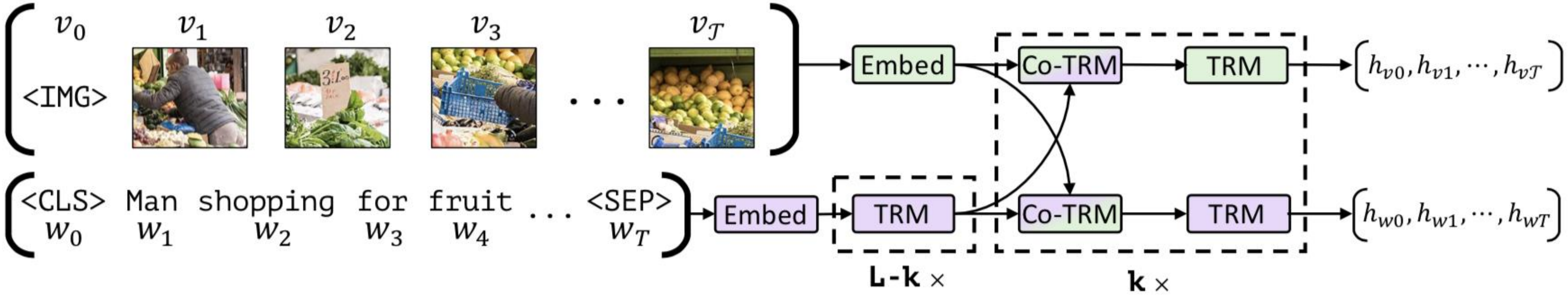
Figure 1. Illustration of the proposed framework.

Related work

ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks

Jiasen Lu[1], Dhruv Batra[1,2], Devi Parikh[1,2], Stefan Lee[1,3]
[1]Georgia Institute of Technology, [2]Facebook AI Research, [3]Oregon State University

- ViLBERT (short for Vision-and-Language BERT)
- Extend BERT to jointly represent images and text
- Co-attentional transformer layers

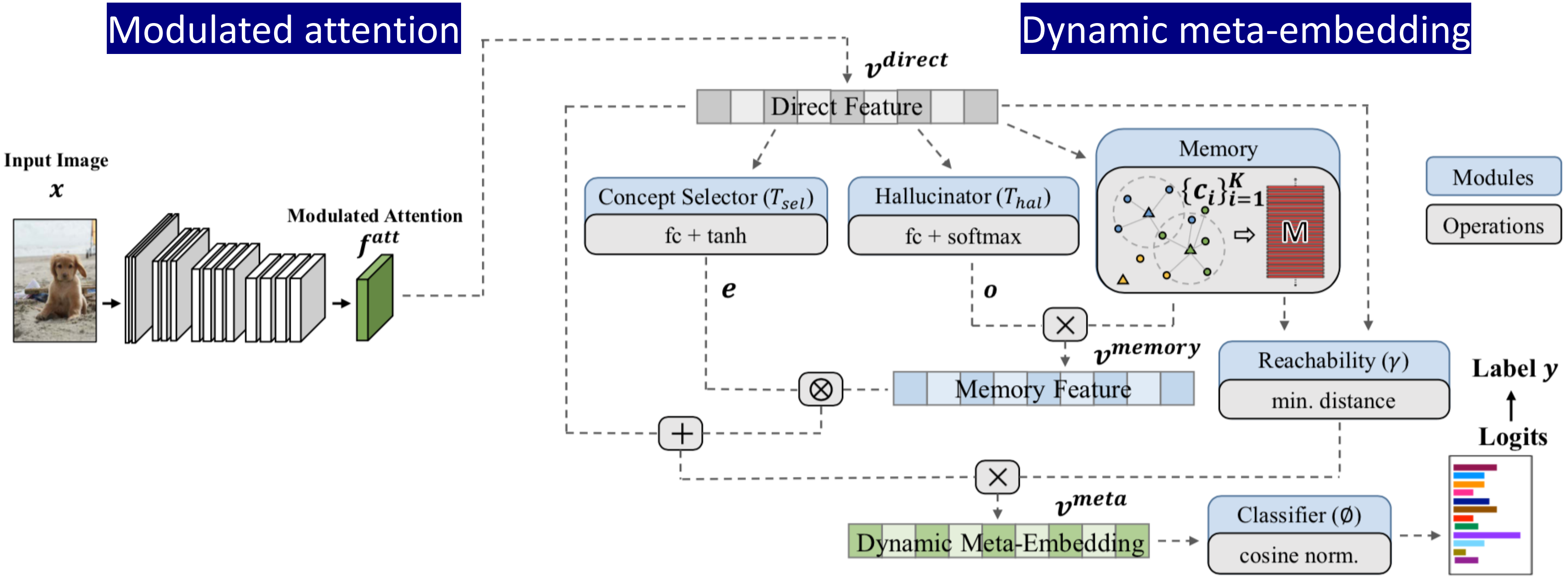


Related work

Large-Scale Long-Tailed Recognition in an Open World

Ziwei Liu[1,2*] Zhongqi Miao[2*] Xiaohang Zhan[1] Jiayun Wang[2] Boqing Gong[2†] Stella X. Yu[2]
[1] The Chinese University of Hong Kong [2] UC Berkeley / ICSI
{zwliu,zx017}@ie.cuhk.edu.hk, {zhongqi.miao,peterwg,stellayu}@berkeley.edu, bgong@outlook.com

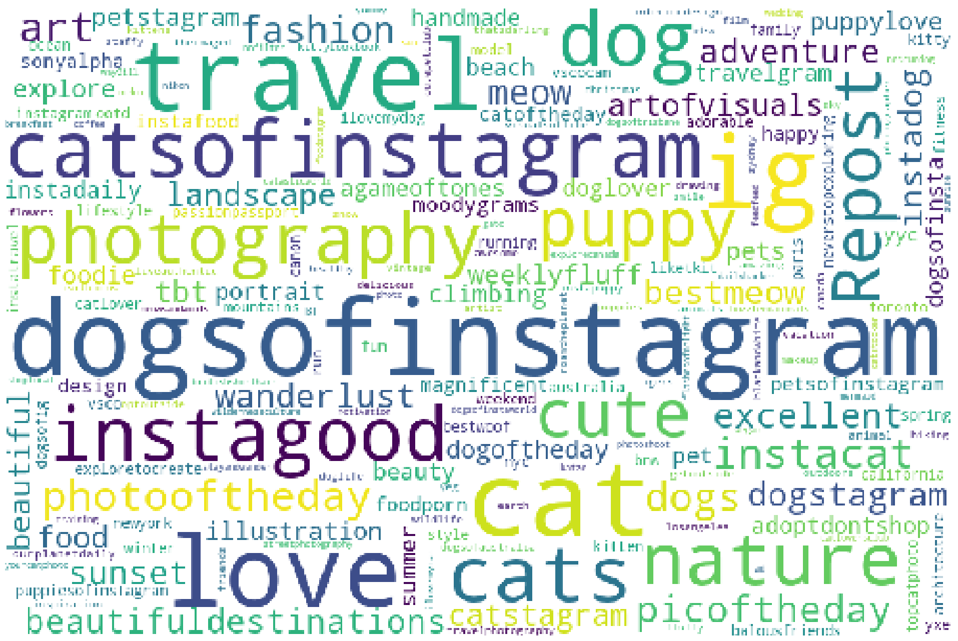
- 2019 CVPR
- **OLTR** (Open Long-Tailed Recognition): Handle imbalanced classification, few-shot learning, and open-set recognition in one integrated algorithm

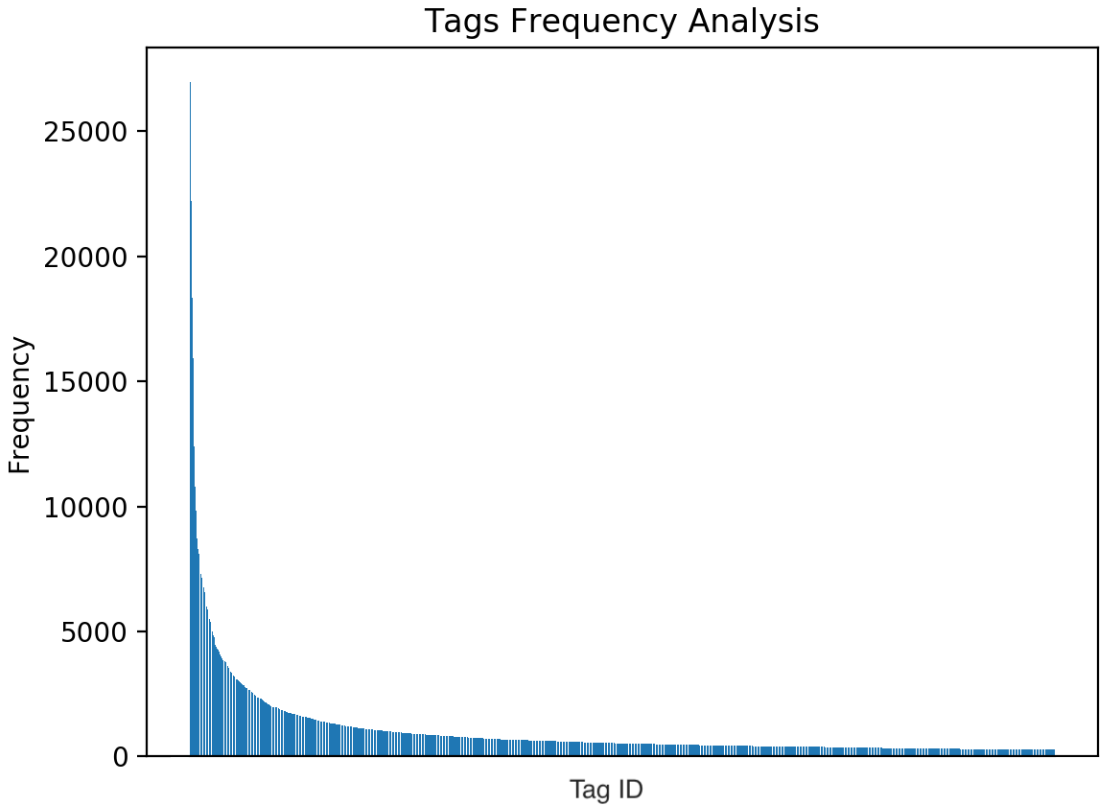


My Work

Dataset: MaCon

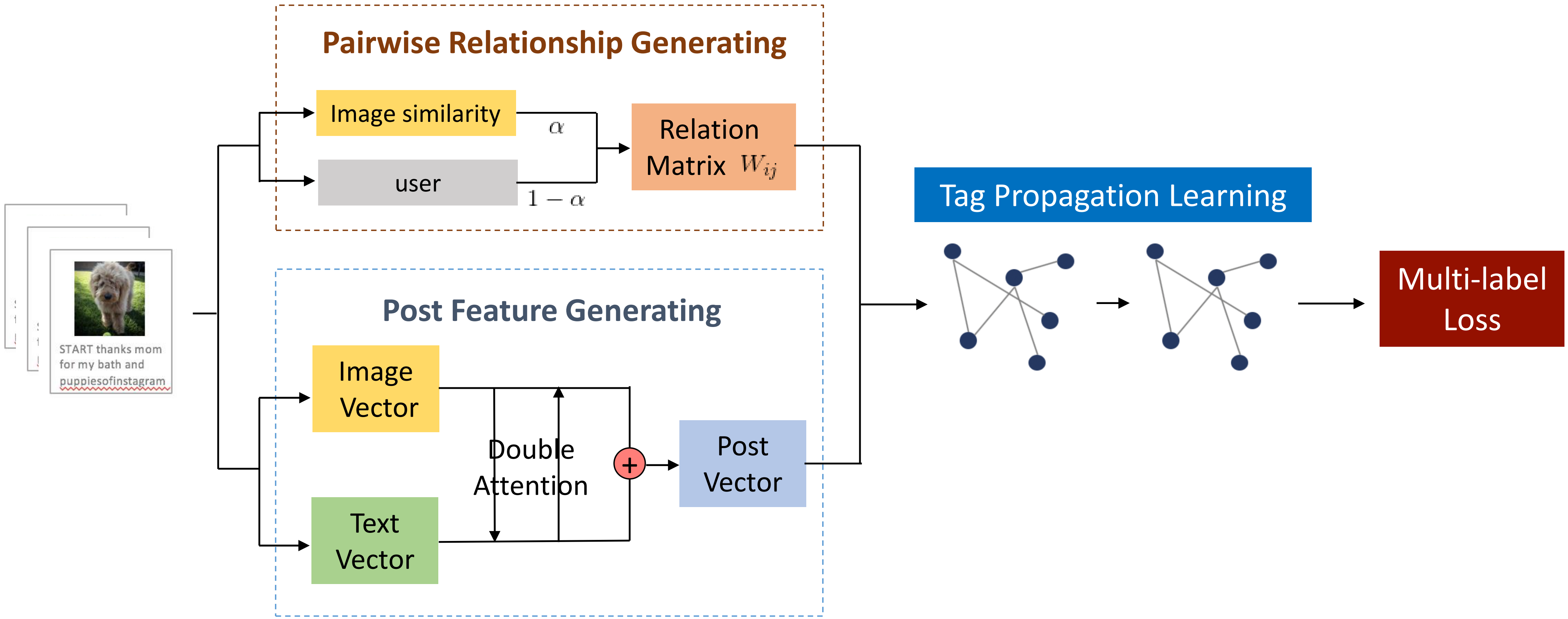
- Analysis of dataset
  - According to hashtag frequency





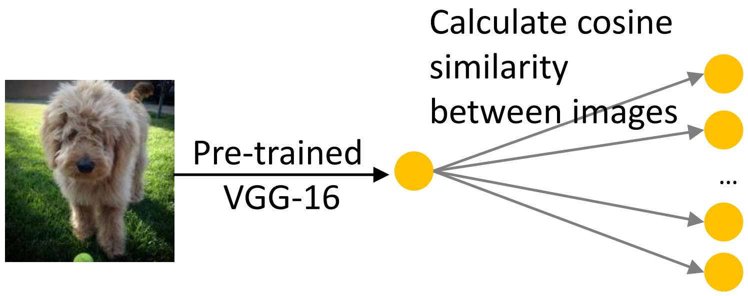
3. The Proposed Approach

3.1 Model Overview



3. The Proposed Approach

3.2 Pairwise Relationship Generating



Relation map of image:

$$I_{ij} = \begin{cases} 1 & \text{,if } i \neq j \text{ and } similarity(img_i, img_j) > \tau \\ 0 & \text{,otherwise} \end{cases}$$

We use the threshold τ to filter image-pairs which have low simiarity.

Relation map of user:

$$U_{ij} = \begin{cases} 1 & \text{,if } i \neq j \text{ and the user of } post_i \text{ and } post_j \text{ are the same} \\ 0 & \text{,otherwise} \end{cases}$$

Relation Matrix

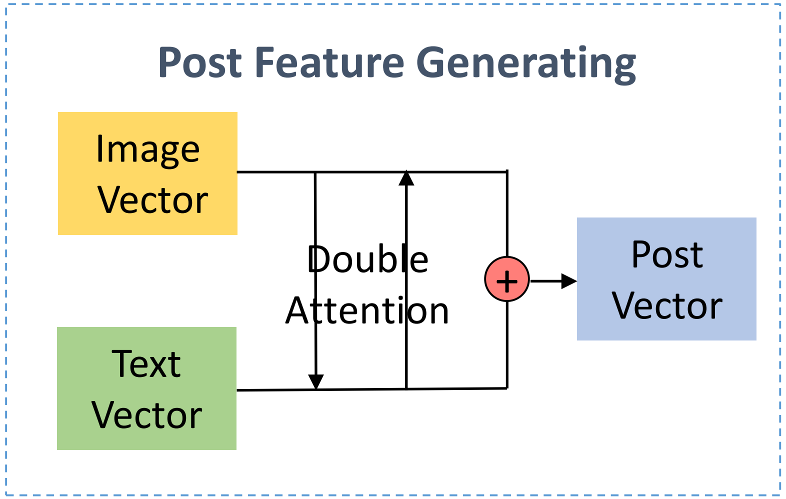
α=1 has the best performance

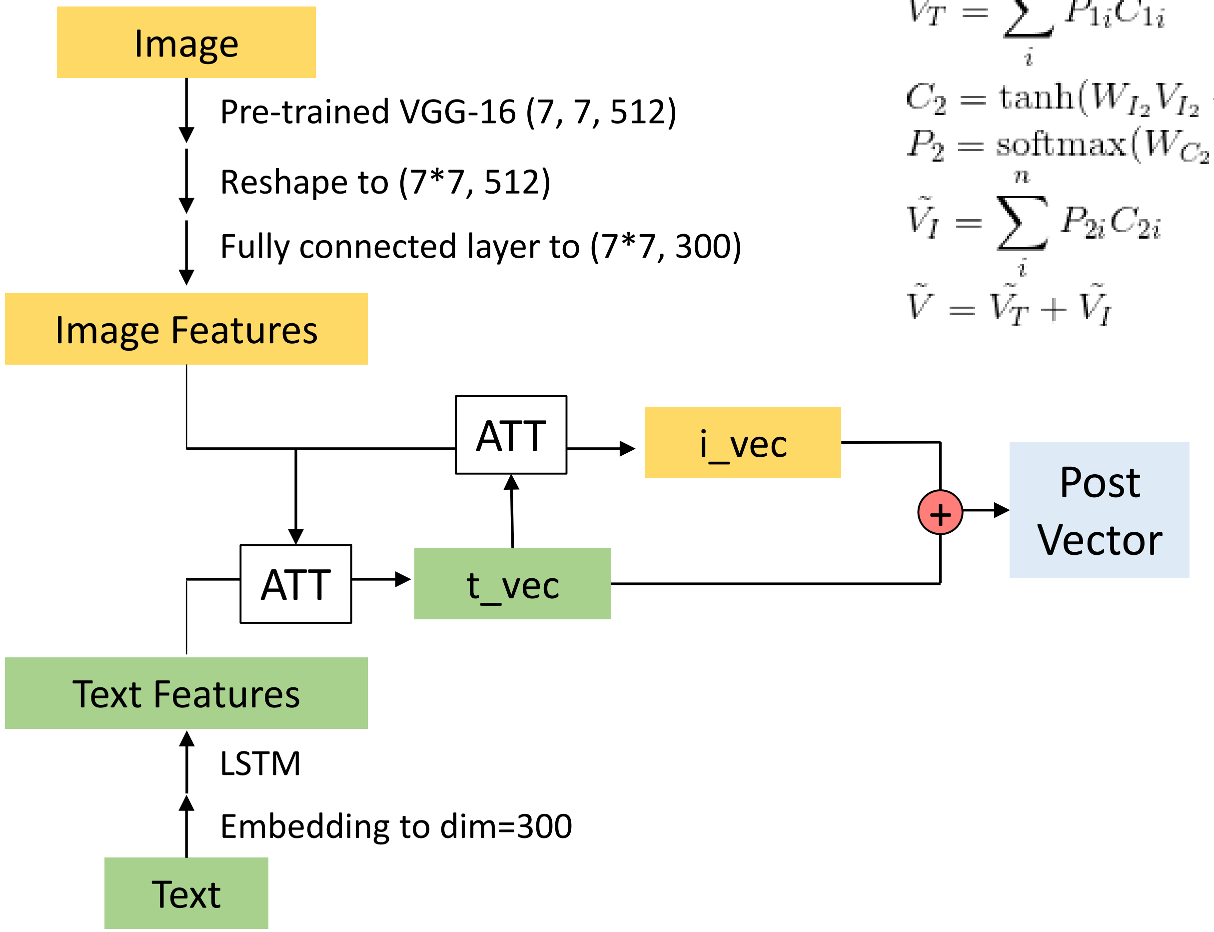
When alpha become close to 1, it seems to consider more about the relation between images.

$$W_{ij} = \begin{cases} \alpha I_{ij} + (1-\alpha)U_{ij} & \text{, if } i \neq j \\ 0 & \text{, otherwise} \end{cases}$$

3. The Proposed Approach

3.3 Post Feature Generating



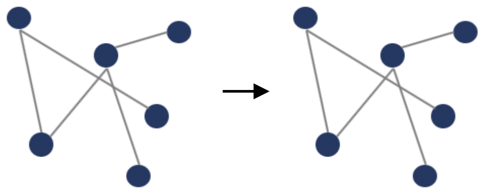


$$C_1 = \tanh(W_I V_I + W_T V_T)$$
$$P_1 = \mathrm{softmax}(W_{C_1} C_1 + b_1)$$
$$\tilde{V_T} = \sum_i^n P_{1i} C_{1i}$$
$$C_2 = \tanh(W_{I_2} V_{I_2} + W_{T_2} \tilde{V_T})$$
$$P_2 = \mathrm{softmax}(W_{C_2} C_2 + b_2)$$
$$\tilde{V_I} = \sum_i^n P_{2i} C_{2i}$$
$$\tilde{V} = \tilde{V_T} + \tilde{V_I}$$

3. The Proposed Approach

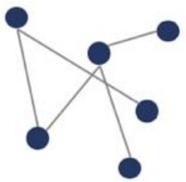
3.4 Tag Propagation Learning

Tag Propagation Learning



GCN

Input



$\boldsymbol{H}^l \in \mathbb{R}^{n\times d}$ (feature matrix)

$\boldsymbol{A} \in \mathbb{R}^{n\times n}$ (adjacency matrix)

Dropout

GCN Layer 1

ReLU

Dropout

GCN Layer 2

Multi-label Loss

After employing the convolutional operation:

$$\boldsymbol{H}^{l+1} = h(\widehat{\boldsymbol{A}}\boldsymbol{H}^l\boldsymbol{W}^l)$$

$\widehat{\boldsymbol{A}} \in \mathbb{R}^{n\times n}$ the normalized version of correlation matrix A.

$\boldsymbol{W}^l \in \mathbb{R}^{d\times d'}$ transformation matrix to be learned.

h(): Non-linear operation. (ex. ReLU)

3. The Proposed Approach

3.5 Training

- The training objective function:

Multi-label Loss

the softmax probability of choosing tag z for input post pi

$$J = \frac{1}{|S|} \sum_{(p_i, \mathcal{T}_i) \in S} \sum_{z \in \mathcal{T}_i} -\log P(z|p_i),$$

a hashtag in the hashtag set

training set

a post and its corresponding hashtag set

4. Experiments

4.1 Evaluation Metrics

Precision(P)	Recall(R)	F1-score(F1)

- Recall@K: The recall value while K candidate hashtags are recommended for each posts.
- Generally, Recall(R) is relatively more important for this performance evaluation.

4.2 Implementation Details

- Implementation: Keras
  - Optimizer = sgd
  - Epochs = 200~300
  - Batch_size = #nodes
- Training : Testing = 9:1

4. Experiments

4.3 Dataset

- MaCon (Zhang et al. 2019):
  - Every post has some attributes such as post_id, words, hashtags, user_id, images.

#Posts	#Users	#Hashtags	Ave_p	Ave_h
624,520	7,497	3896	83.3	6.41

- Sub-dataset that is used for the following experiments

	Sub-1	Sub-2	Sub-3
Node number	11,607	25,259	58,665
Edge Number	68,029	165,392	165,238
Tag Frequency	Top 50	Top 100	Tag 200
Length of Tags per posts	7~10	7~10	5~8

4. Experiments

4.4 Experimental Results

4.4.1 Comparisons with State-of-the-Arts

Method	(Size of dataset: 11,607)			(Size of dataset: 25,259)			(Size of dataset: 58,665)		
	P @10	R @10	F1 @10	P @10	R @10	F1 @10	P @10	R @10	F1 @10
1-layer DNN (image + text)	0.326	0.409	0.362	0.439	0.537	0.481	TBD	TBD	TBD
Co-Attention (CoA)	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD
MaCon (ATT + user habit)	0.325	0.413	0.363	0.218	0.267	0.239	0.103	0.168	0.127
ATT (my ATT) + GCN	0.357	0.448	0.396	0.453	0.554	0.496	0.259	0.416	0.317

- 1-layer DNN: Word embedding + LSTM + DNN
- Co-Attention(CoA) [Zhang et al.2017]
- MaCon [Zhang et al. 2019]

4. Experiments

4.4 Experimental Results

4.4.2 Ablation Studies

Effects of Attention and GCN Module

Method	(Size of dataset: 11,607 posts)		
	P @10	R @10	F1 @10
GCN only	0.328	0.409	0.363
ATT only	0.289	0.361	0.320
ATT (my ATT) + GCN	0.357	0.448	0.396

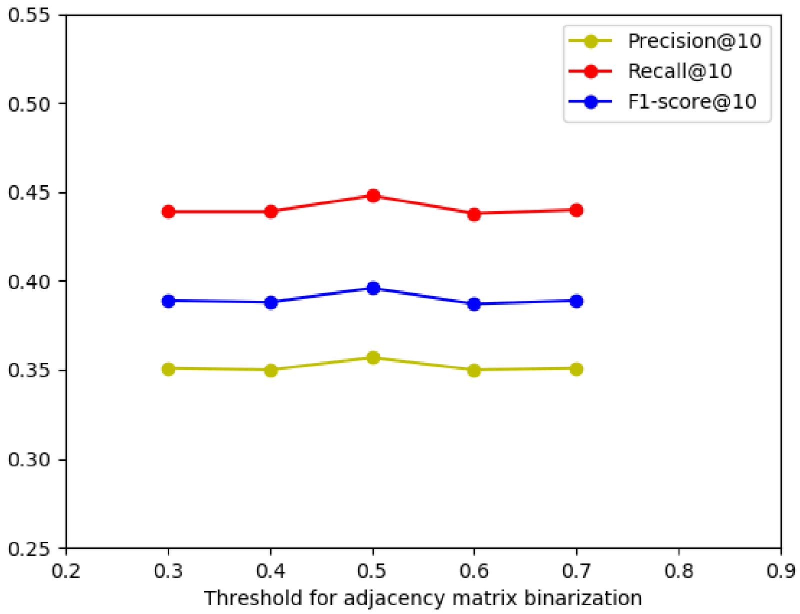
4. Experiments

4.4 Experimental Results

4.4.2 Ablation Studies

Effects of different threshold value τ
(in calculating image similarity for adjacency matrix binarization)

Threshold τ	(Size of dataset: 11,607 posts) P @10	R @10	F1 @10
0.3	0.351	0.439	0.389
0.4	0.350	0.439	0.388
0.5	0.357	0.448	0.396
0.6	0.350	0.438	0.387
0.7	0.351	0.440	0.389



4. Experiments

4.4 Experimental Results

4.4.2 Ablation Studies

α=1 has the best performance

Effects of different α for final relation matrix

$$W_{ij} = \begin{cases} \alpha I_{ij} + (1-\alpha)U_{ij} & , \text{if } i \neq j \\ 0 & , \text{otherwise} \end{cases}$$

$$W_{ij} = \begin{cases} \alpha I_{ij} + (1-\alpha)T_{ij} & , \text{if } i \neq j \\ 0 & , \text{otherwise} \end{cases}$$

[Adding user information]

α	(Size of dataset: 11,607 posts)		
	P @10	R @10	F1 @10
0.5	0.348	0.436	0.386
0.9	0.353	0.443	0.391
1	0.357	0.448	0.396

[Adding word information]

α	(Size of dataset: 11,607 posts)		
	P @10	R @10	F1 @10
0.5	0.350	0.438	0.388
0.8	0.351	0.440	0.389
1	0.357	0.448	0.396

References

- Kipf & Welling (ICLR 2017), Semi-Supervised Classification with Graph Convolutional Networks
- Defferrard et al. (NIPS 2016), Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering
- https://slideplayer.com/slide/7806012/
- https://towardsdatascience.com/mapping-word-embeddings-with-word2vec-99a799dc9695
- http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
- https://www.tensorflow.org/tutorials/representation/word2vec
- B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online Learning of Social Representations," *KDD*, 2014
- A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," *KDD*, 2016
- K. Tu, R. Cui, X. Wang, P. S. Yu, and W. Zhu, "Deep Recursive Network Embedding with Regular Equivalence," *KDD*, 2018