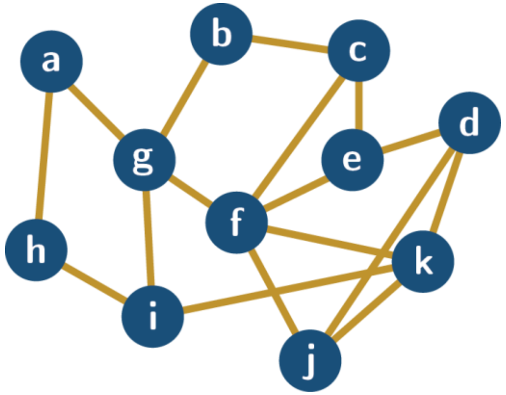


RESEARCH DISCUSSION:

NETWORK EMBEDDING

Paper title: *DeepWalk: Online Learning of Social Representations,*

Bryan Perozzi, Rami Al-Rfou, Steven Skienna, Stony Brook University

KDD'14, August 24-27, 2014, New York, NY, USA.

Presenter: Hoang Nguyen
Tokyo Institute of Technology

@ University of Tokyo – 15th March 2016

CONTENT

- Idea: NLP technique applied to Network Embedding
- Problem addressing
- Building blocks of the system
- Authors' experiments and evaluation
- Some thought on the paper

CONTENT

- Idea: NLP technique applied to Network Embedding
- Problem addressing
- Building blocks of the system
- Authors' experiments and evaluation
- Some thought on the paper

CONTENT

- Idea: NLP technique applied to Network Embedding
- Problem addressing
- Building blocks of the system
- Authors' experiments and evaluation
- Some thought on the paper

CONTENT

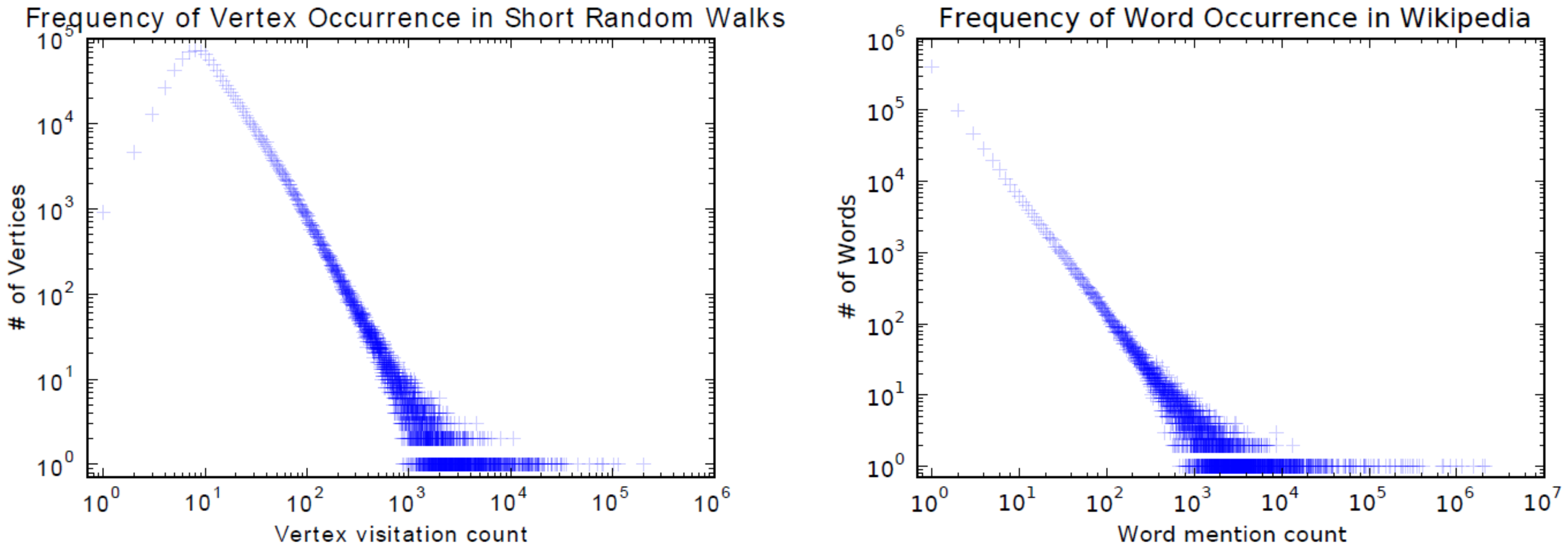
- Idea: NLP technique applied to Network Embedding
- Problem addressing
- Building blocks of the system
- Authors' experiments and evaluation
- Some thought on the paper

CONTENT

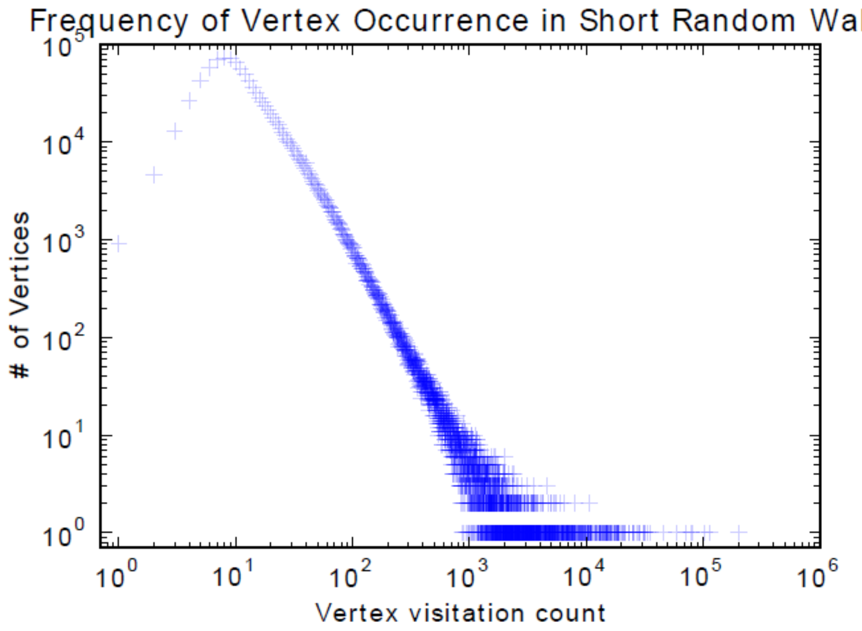
- Idea: NLP technique applied to Network Embedding
- Problem addressing
- Building blocks of the system
- Authors' experiments and evaluation
- Some thought on the paper

- Idea: NLP technique applied to Network Embedding
- Problem addressing
- Building blocks of the system
- Authors' experiments and evaluation
- Some thought on the paper

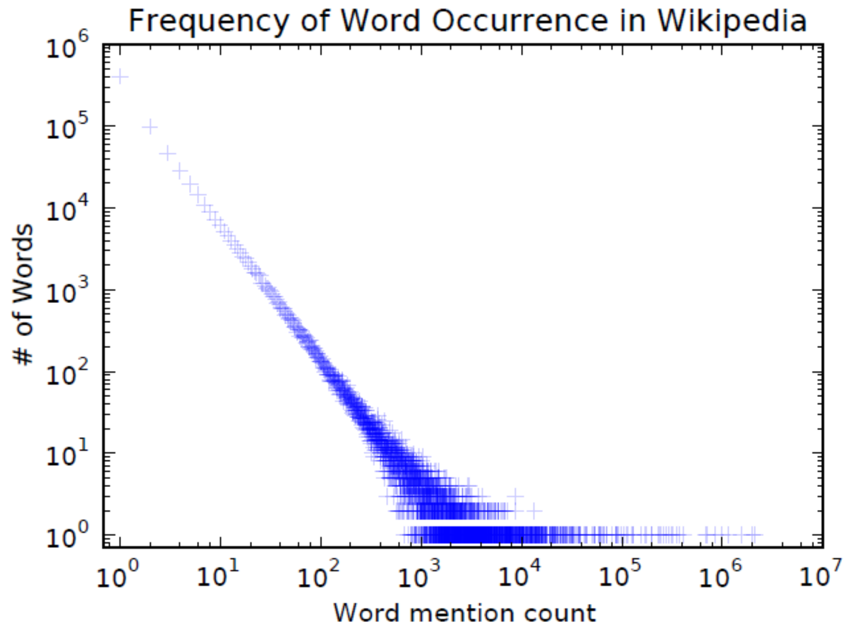
NLP techniques applied to Network Embedding



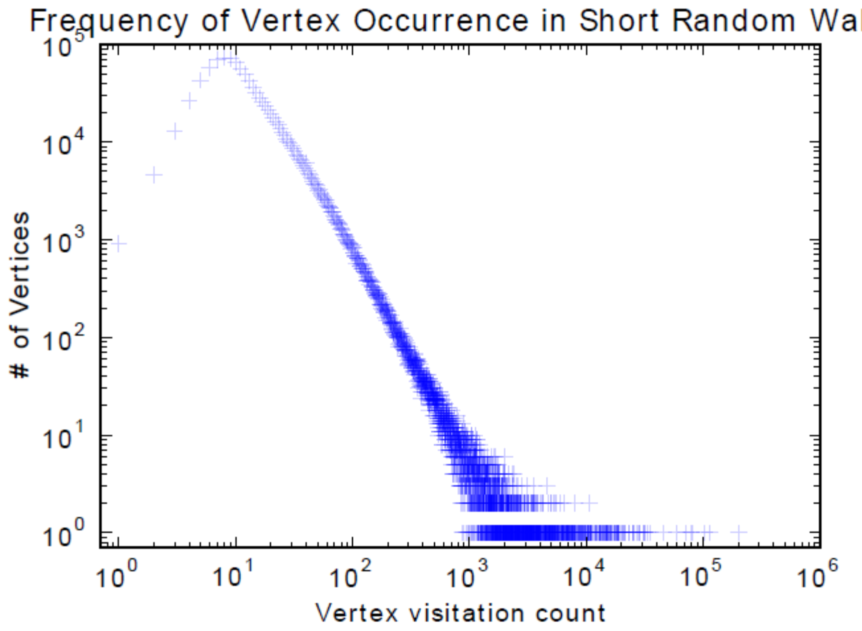
NLP techniques applied to Network Embedding



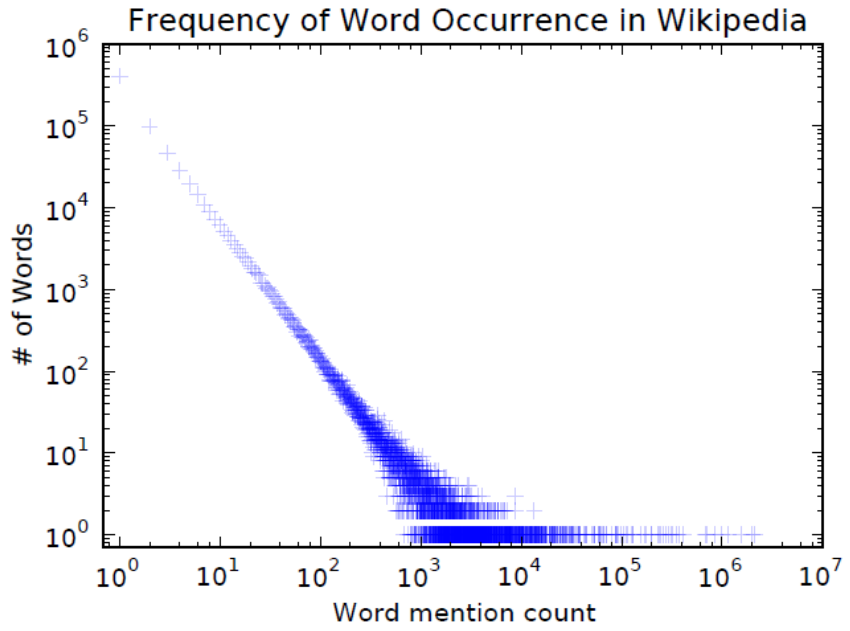
YouTube Social **Graph**



NLP techniques applied to Network Embedding

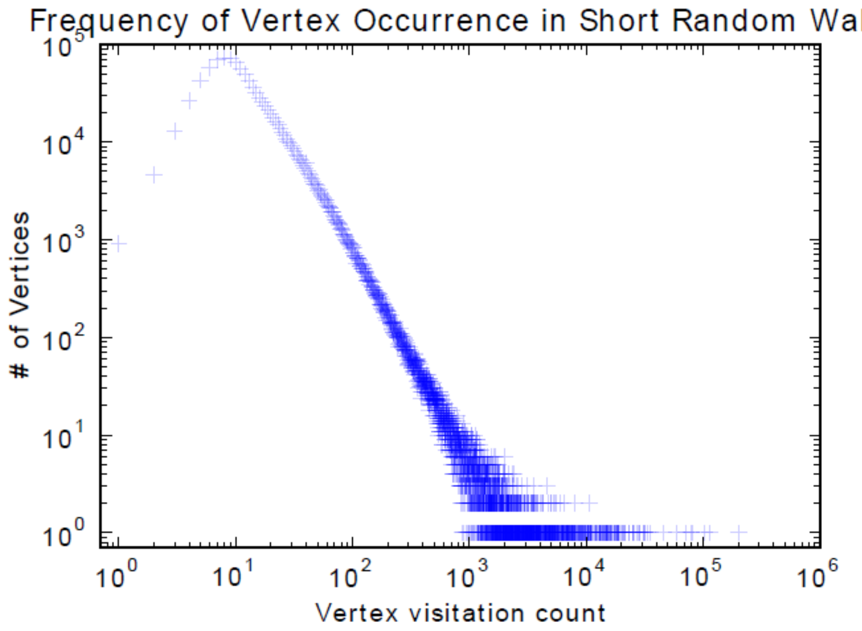


YouTube Social **Graph**

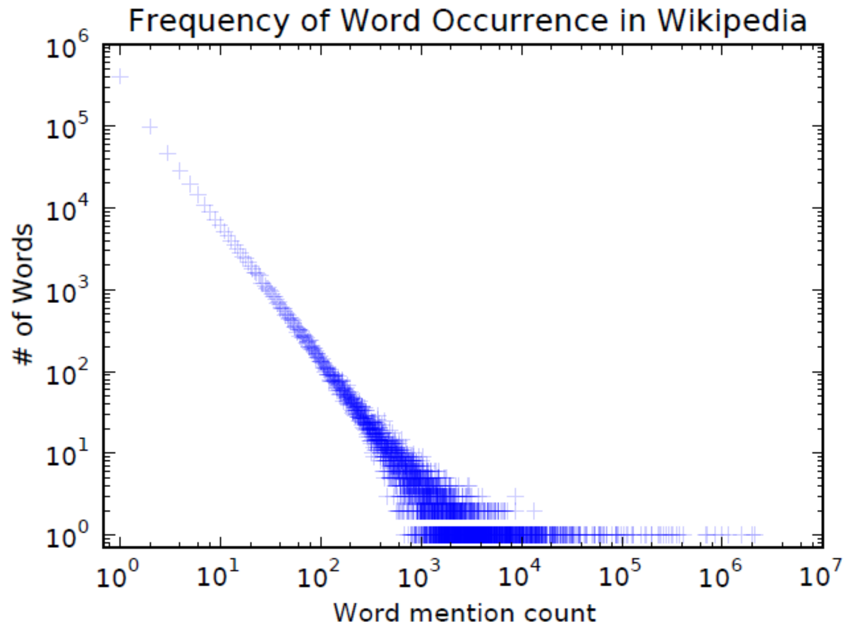


Wikipedia Article **Text**

NLP techniques applied to Network Embedding



YouTube Social **Graph**



Wikipedia Article **Text**

* The resemblance in Power Law distribution inspired the authors to apply NLP to graph!

NLP techniques applied to Network Embedding

$\boxed{\text{The brown fox}}$

$(v_1, v_2,\ v_3, \ldots,\ v_{i-1})$

TEXT

NLP techniques applied to Network Embedding

$\boxed{\text{The brown fox}}$ jumped

$(v_1, v_2, v_3, \dots, v_{i-1})$

$$\left.\begin{array}{l}\text{jumped}\\ \text{kicked}\\ \text{ate}\\ \text{home}\\ \text{lazy}\\ \dots\end{array}\right\} v_i$$

TEXT

NLP techniques applied to Network Embedding

$\boxed{\text{The brown fox}}$ jumped

$(v_1, v_2, v_3, \ldots, v_{i-1})$

$$\left.\begin{array}{l}\text{jumped}\\ \text{kicked}\\ \text{ate}\\ \text{home}\\ \text{lazy}\\ \ldots\end{array}\right\} v_i$$

$$\Pr\left(v_i \mid (v_1, v_2, \ldots, v_{i-1}) \right)$$

TEXT

NLP techniques applied to Network Embedding

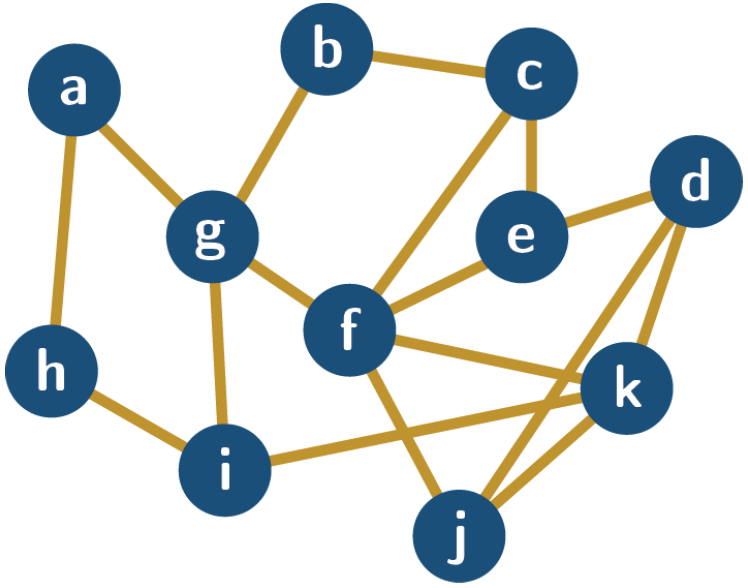
$\boxed{\text{The brown fox}}$ $\left.\begin{array}{l}\text{jumped}\\ \text{kicked}\\ \text{ate}\\ \text{home}\\ \text{lazy}\\ \dots\end{array}\right\} v_i$

$(v_1, v_2, v_3, \dots, v_{i-1})$

$$\Pr\left(v_i \mid \left(\Phi(v_1), \Phi(v_2), \dots, \Phi(v_{i-1})\right)\right)$$

TEXT

NLP techniques applied to Network Embedding



The brown fox

$(v_1, v_2, v_3, \ldots, v_{i-1})$

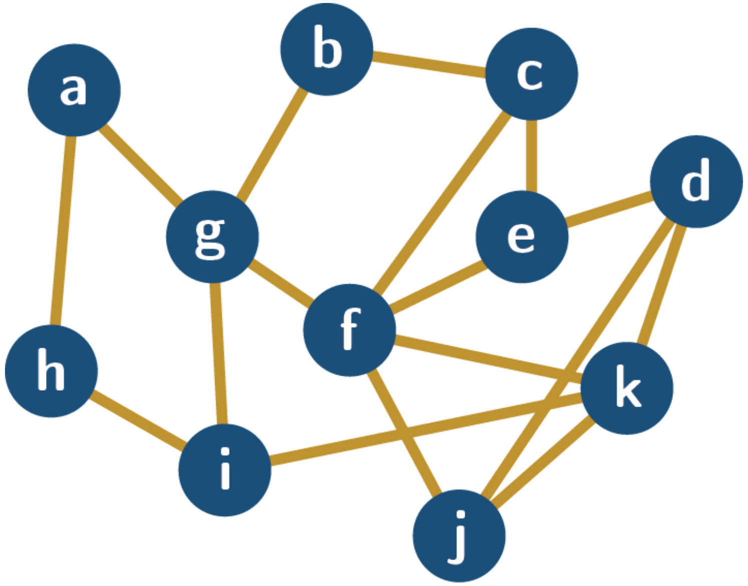
jumped
kicked
ate
home
lazy
...

v_i

$$\Pr\left(v_i \mid (\Phi(v_1), \Phi(v_2), \ldots, \Phi(v_{i-1}))\right)$$

GRAPH **TEXT**

NLP techniques applied to Network Embedding



$\boxed{\text{The brown fox}}$ jumped kicked ate home lazy …

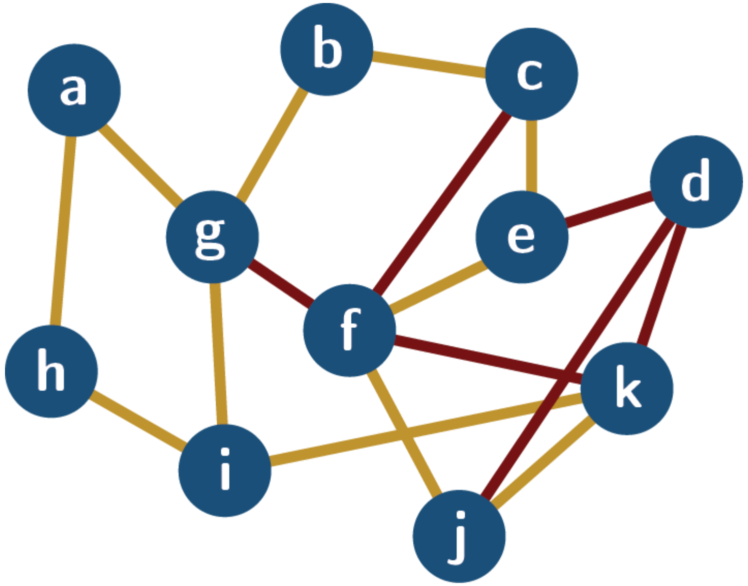
$$\underbrace{(v_1, v_2, v_3, \ldots, v_{i-1})}\quad \left.\begin{matrix}\text{jumped}\\ \text{kicked}\\ \text{ate}\\ \text{home}\\ \text{lazy}\\ \ldots\end{matrix}\right\} v_i$$

How to create a "context" ?

$$\mathbf{Pr}\left(v_i \middle| \left(\Phi(v_1), \Phi(v_2), \ldots, \Phi(v_{i-1})\right) \right)$$

GRAPH **TEXT**

NLP techniques applied to Network Embedding



Random Walk: e > d > k > f > g >
c > f > k > d > j

The brown fox jumped / kicked / ate / home / lazy / …

$(v_1, v_2, v_3, \dots, v_{i-1})$ $\quad v_i$

$\mathbf{Pr}\left(v_i \mid (\Phi(v_1), \Phi(v_2), \dots, \Phi(v_{i-1})\right)$

GRAPH **TEXT**

- Idea: NLP technique applied to Network Embedding
- Problem addressing
- Building blocks of the system
- Authors' experiments and evaluation
- Some thought on the paper

Problem addressing

$G_L(V, E, X, Y)$ - Labeled graph with vector representations X and labels Y.

Problem addressing

$\boldsymbol{G_L(V, E, X, Y)}$ - Labeled graph with vector representations X and labels Y.

$\boldsymbol{X}$ - $|\mathrm{V}| \times \boldsymbol{d}$ matrix stores $\boldsymbol{d}$-features vectors for each vertex.

Problem addressing

$G_L(V, E, X, Y)$ - Labeled graph with vector representations X and labels Y.

X - $|V| \times d$ matrix stores d-features vectors for each vertex.

Y - $|V| \times l$ matrix stores labels for each vertex. There are l labels.

Problem addressing

$G_L(V, E, X, Y)$ - Labeled graph with vector representations X and labels Y.

X - $|\mathrm{V}| \times d$ matrix stores d-features vectors for each vertex.

Y - $|\mathrm{V}| \times l$ matrix stores labels for each vertex. There are l labels.

Objective: Learn a hypothesis **H** that maps elements of X to the label set.

$$H: X \xrightarrow{G} Y$$

Problem addressing

$G_L(V, E, X, Y)$ - Labeled graph with vector representations X and labels Y.

X - $|V| \times d$ matrix stores d-features vectors for each vertex.

Y - $|V| \times l$ matrix stores labels for each vertex. There are l labels.

Objective: Learn a hypothesis **H** that maps elements of X to the label set.

$$H : X \xrightarrow{G} Y$$

Problem: Cascading error when integrating structural representation with labeling.

$\Rightarrow$ separate the 2 task!

Problem addressing

Objective of the paper: Learn the latent representation $\boldsymbol{X_E}$. Guarantee:

Problem addressing

Objective of the paper: Learn the latent representation $\boldsymbol{X_E}$. Guarantee:

- Adaptability
- Community aware
- Low dimensional
- Continuous

- Idea: NLP technique applied to Network Embedding
- Problem addressing
- Building blocks of the system
- Authors' experiments and evaluation
- Some thought on the paper

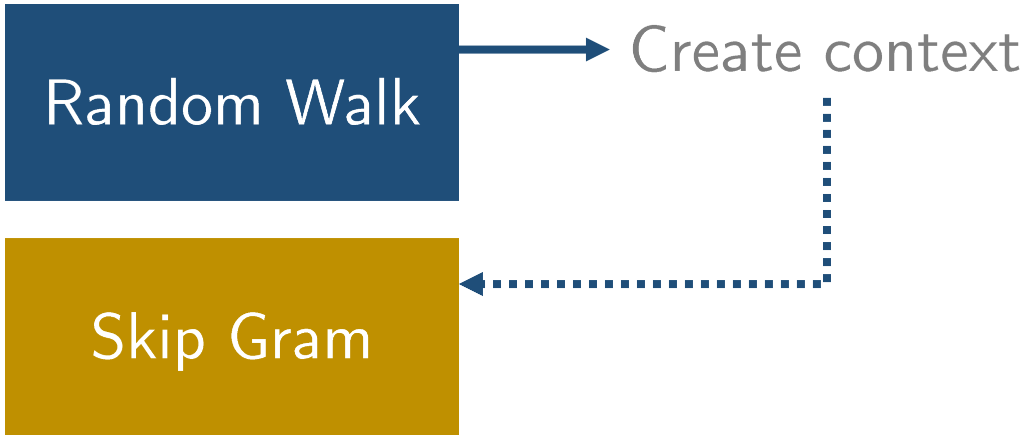
Method

Random Walk

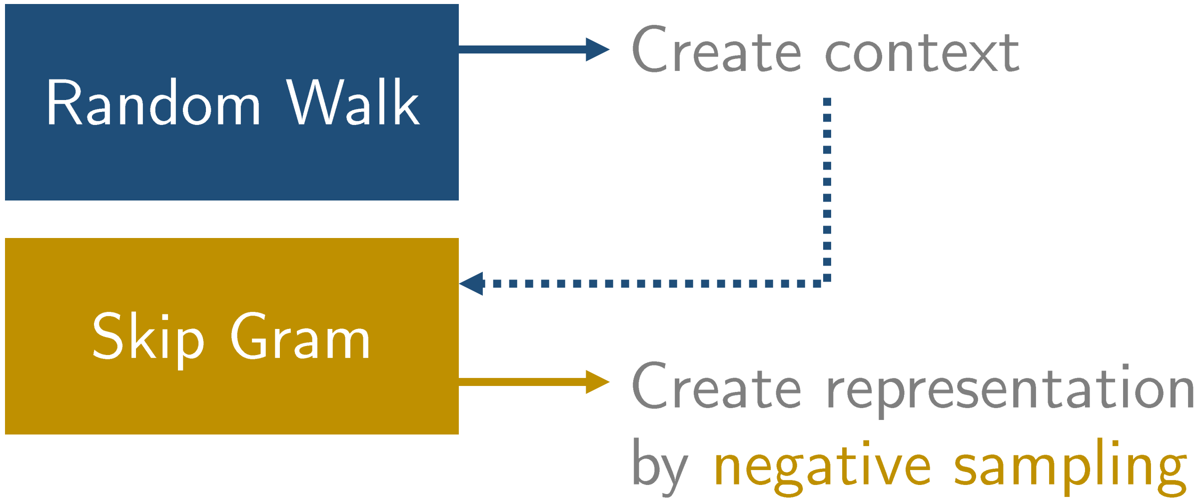
Method

Random Walk → Create context

Method



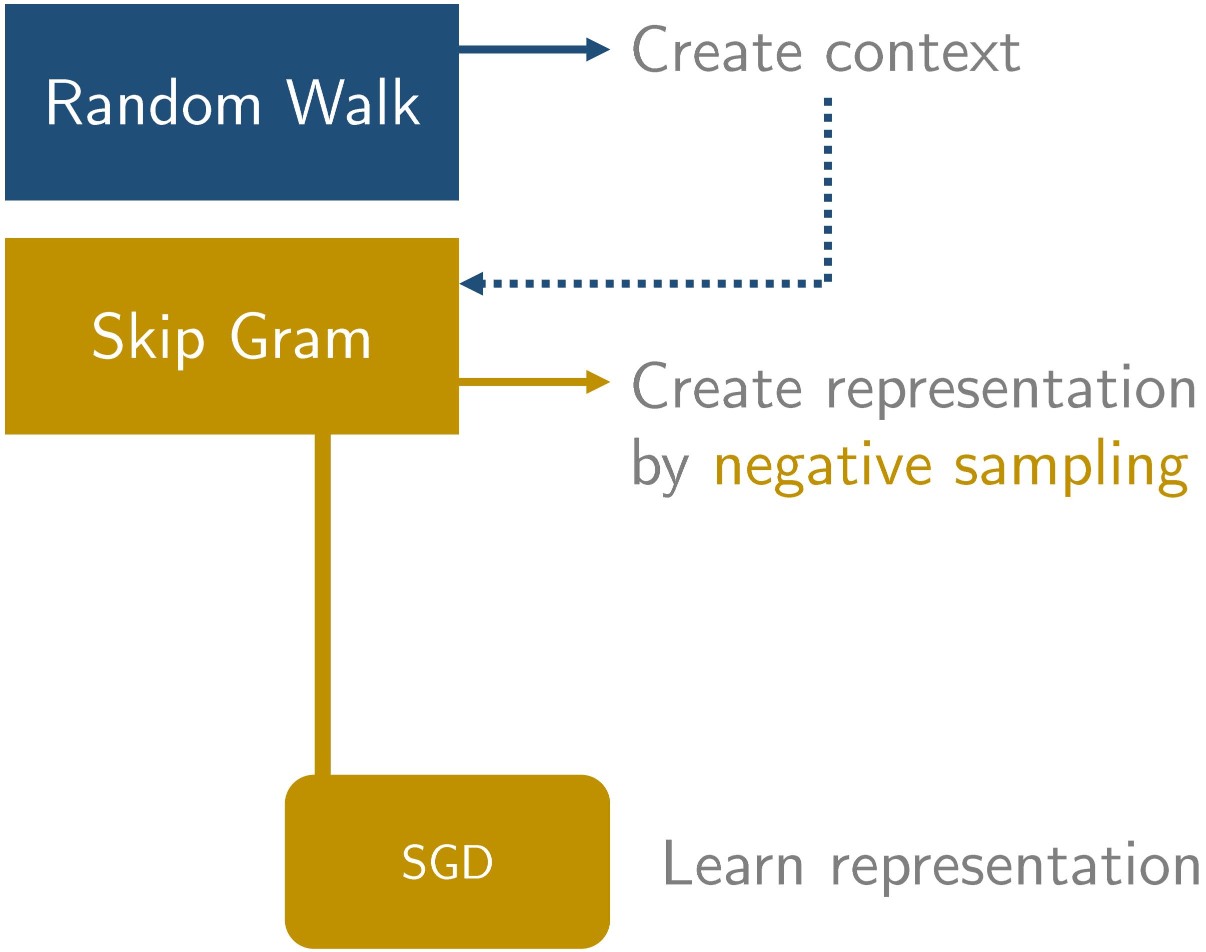
Method


Random Walk
Create context
Skip Gram
Create representation
by negative sampling

Method

Parameters:

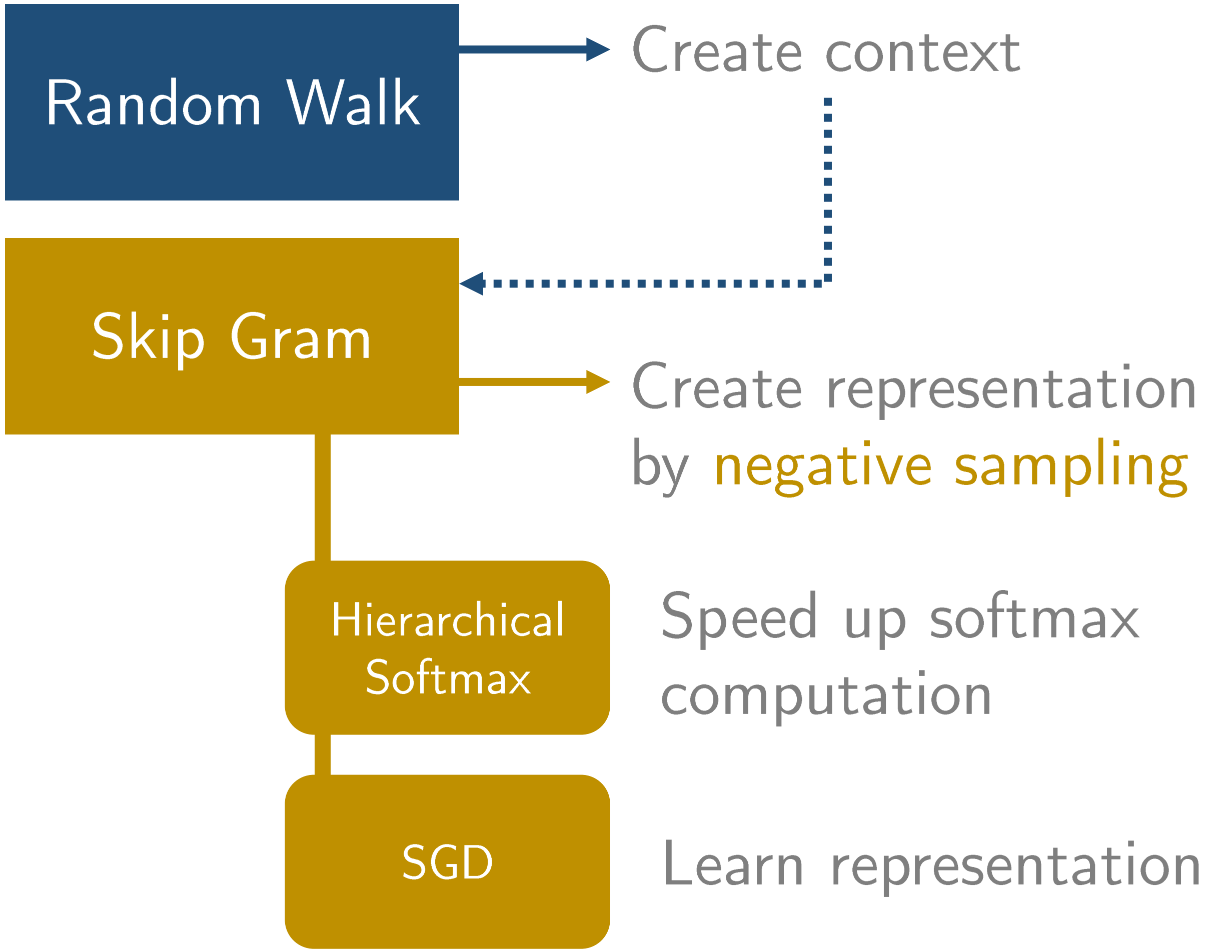
$\boldsymbol{\Phi}$ - $|\mathbf{V}| \times \boldsymbol{d}$ matrix stores latent representation



Method

Parameters:

- $\mathbf{\Phi}$ - $|\mathbf{V}| \times \boldsymbol{d}$ matrix stores latent representation
- $\mathbf{\Psi}$ - $|\mathbf{V}| \times \boldsymbol{d}$ matrix stores hierarchical softmax

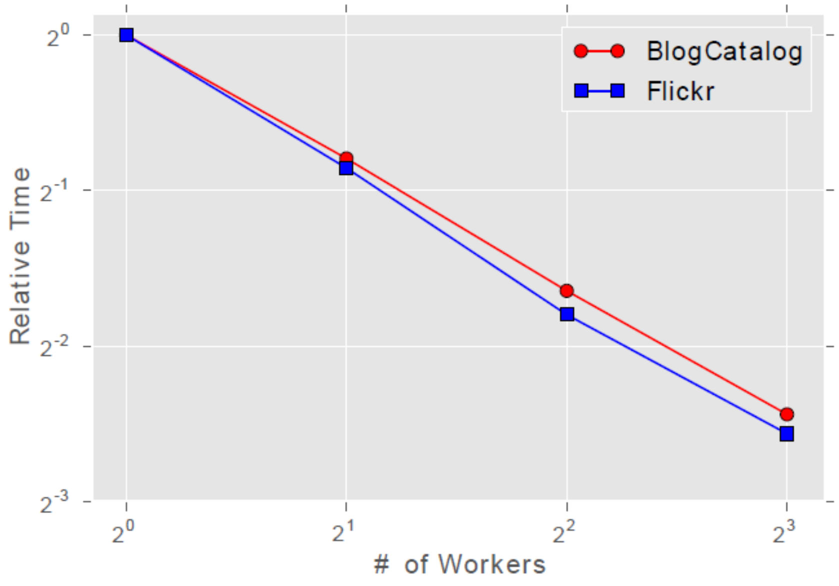


- Idea: NLP technique applied to Network Embedding
- Problem addressing
- Building blocks of the system
- Authors' experiments and evaluation
- Some thought on the paper

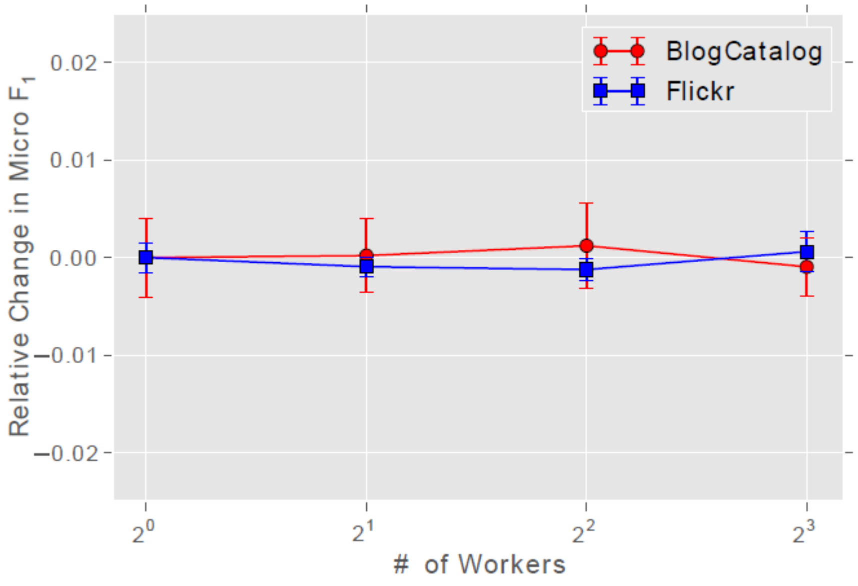
Authors' experiments

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

Authors' experiments



Running Time



Performance

- It is easy to implement for shared-memory system, but shared-nothing system implementation might not be easy.

Authors' experiments

- Repeat the sampling / training process 10 times for each T_R.
- Report the average precision:

Macro-F1

$$\text{Macro-F1} = \frac{\sum P_i}{\sum_i 1}$$

Micro-F1

$$\text{Micro-F1} = \frac{\sum TP_i}{\sum (TP_i + FN_i)}$$

Authors' experiments

	% 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

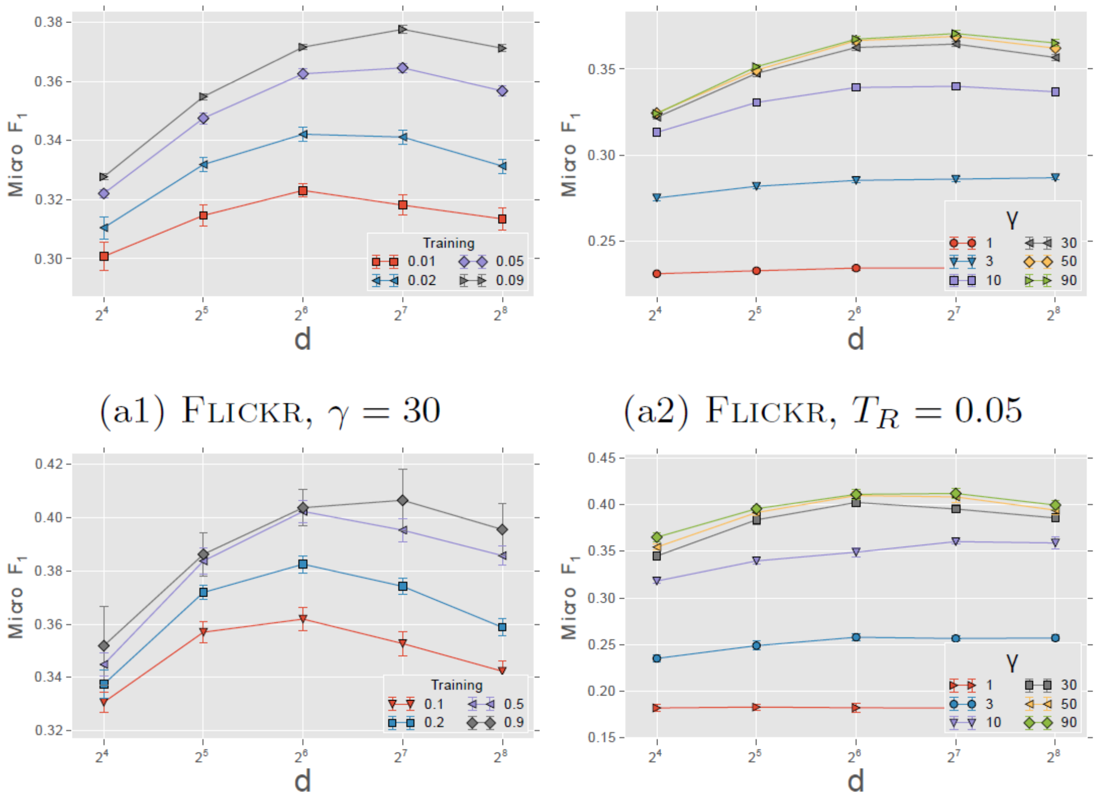
YouTube – 1,138,499 : 2,990,443 : 47

Authors' experiments

	% Labeled Nodes	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1(%)	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
	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
Macro-F1(%)	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
	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

BlogCatalog – 10,312 : 333,983 : 39

Authors' experiments

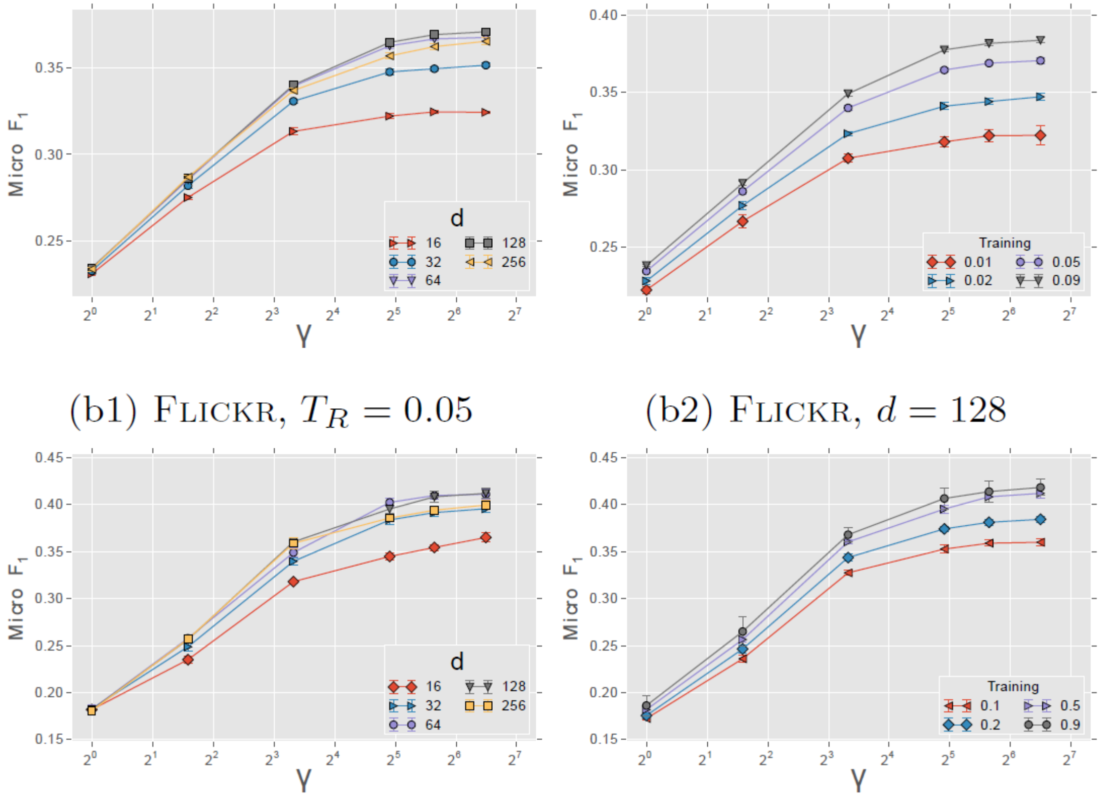


(a1) FLICKR, $\gamma = 30$ (a2) FLICKR, $T_R = 0.05$

(a3) BLOGCATALOG, $\gamma = 30$ (a4) BLOGCATALOG, $T_R = 0.5$

- This results show that a choice of appropriate latent dimension will have significant effect on the system.

Authors' experiments



(b1) FLICKR, $T_R = 0.05$ (b2) FLICKR, $d = 128$

(b3) BLOGCATALOG, $T_R = 0.5$ (b4) BLOGCATALOG, $d = 128$

- Longer walk length is beneficial to the system. However, a good result can be obtained by a relatively small walk length.

- Idea: NLP technique applied to Network Embedding
- Problem addressing
- Building blocks of the system
- Authors' experiments and evaluation
- Some thought on the paper

Conclusion

Contribution **Improvement / Problem**

Conclusion

Contribution

Improvement / Problem

- Idea of applying NPL to Graph embedding.
- Propose using Random Walk to create context in graph.
- Fast algorithm.
- Easy to distribute computing task to different threads.
- Open-source implementation and throughout evaluation of their methods.

Conclusion

Contribution

- Idea of applying NPL to Graph embedding.
- Propose using Random Walk to create context in graph.
- Fast algorithm.
- Easy to distribute computing task to different threads.
- Open-source implementation and throughout evaluation of their methods.

Improvement / Problem

- There is no loss function or competitive analysis of the stochastic process.
- Random Walk, while somewhat similar to Depth First Search, only capture local representation.
- Parallelization is simple on one machine with multiple core CPU, but become unrealistic on fully distributed system..

THANK YOU VERY MUCH
FOR LISTENING