Hierarchical Taxonomy Aware Network Embedding

Jianxin Ma (https://jianxinma.github.io), Peng Cui, Xiao Wang, Wenwu Zhu
Department of Computer Science and Technology, Tsinghua University, Beijing, China

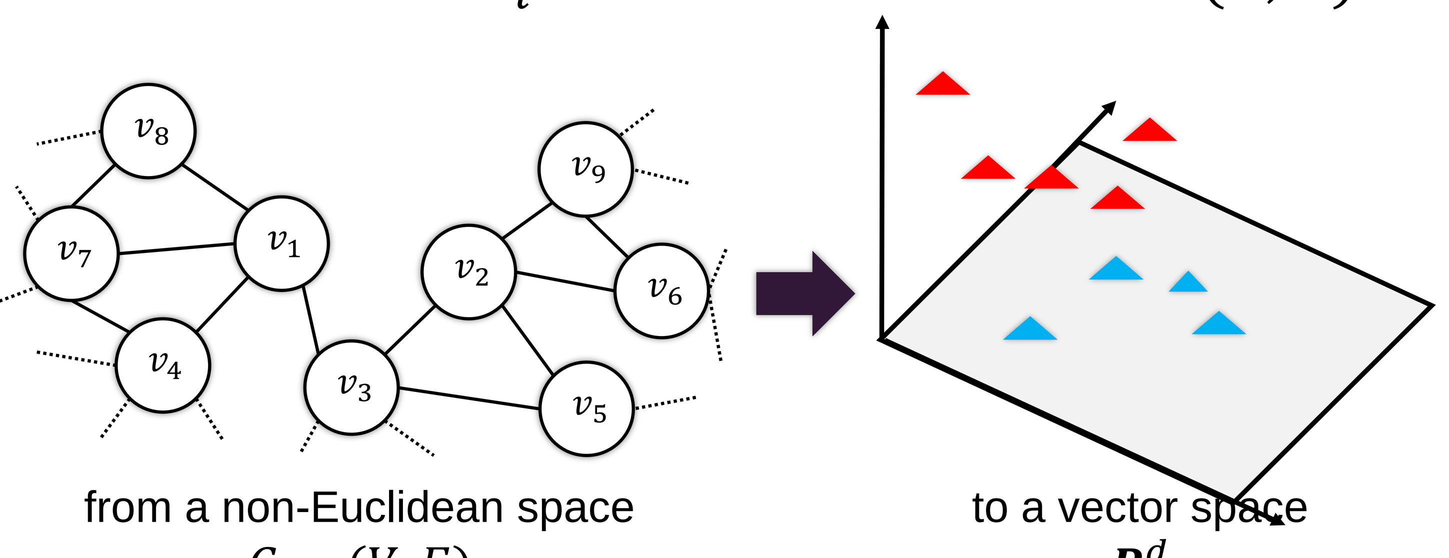
KDD2018

media and network lab

Tsinghua University

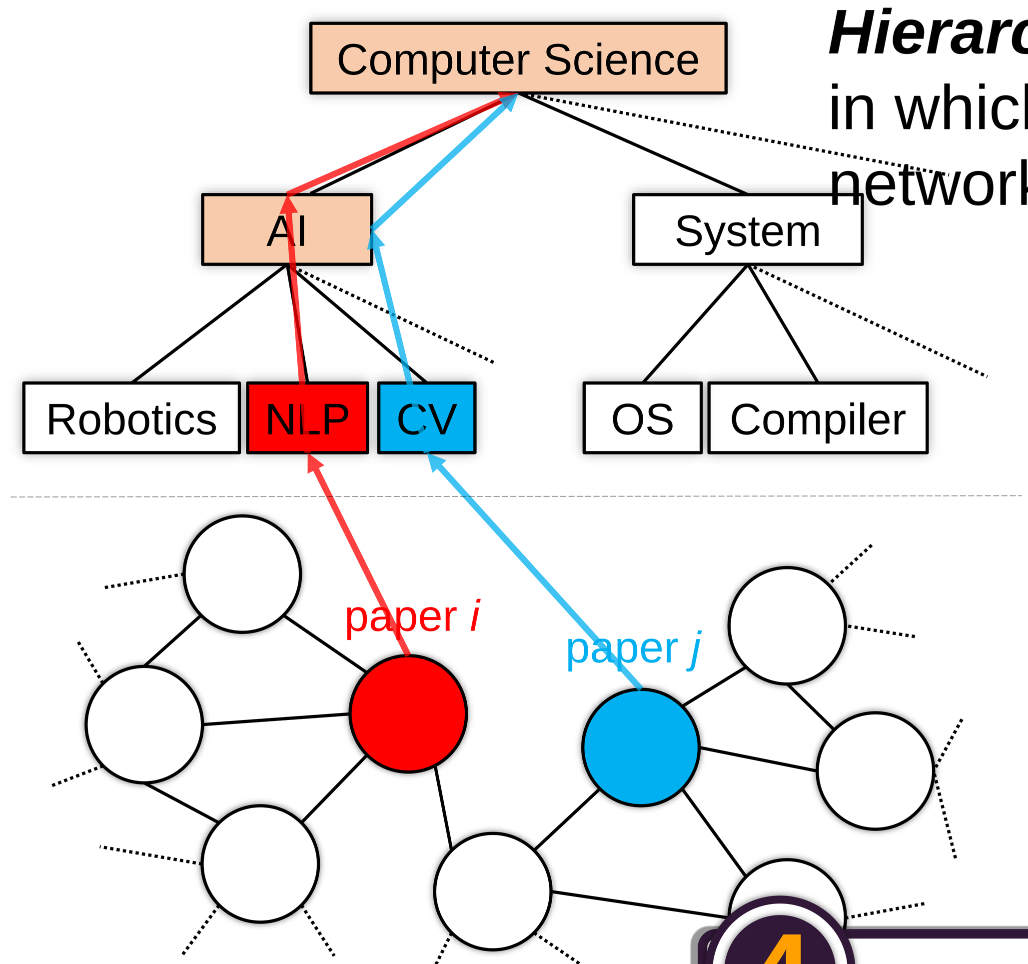
1 (Background) Network Embedding

To learn a d-dimensional vector representation $\boldsymbol{x}_i \in \boldsymbol{R}^d$ for each vertex $v_i \in V$ in the network $G = (V, E)$.



from a non-Euclidean space $G = (V, E)$ → to a vector space $\boldsymbol{R}^d$

2 (Motivation) Hierarchical Taxonomy



Hierarchical taxonomy is a tree structure in which the entities (e.g. papers of a citation network) are classified hierarchically.

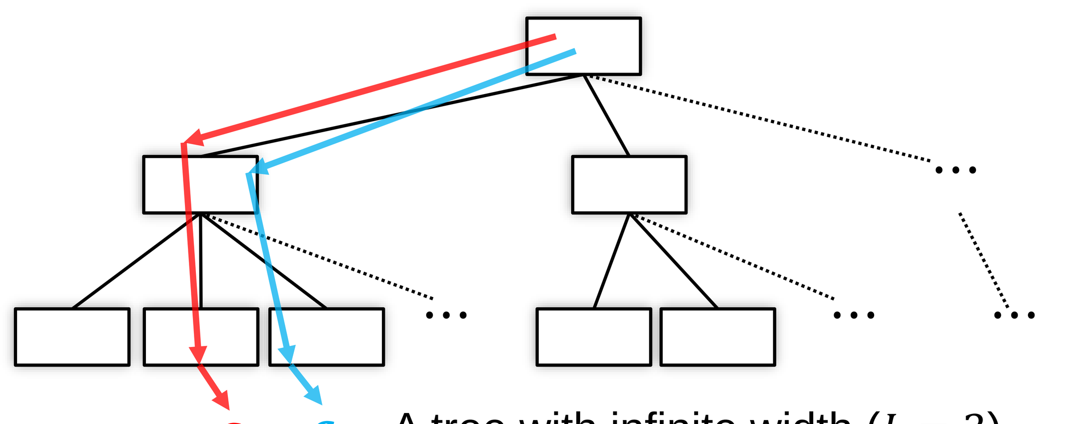
1. Multiple Levels of Granularity

- coarse-grained → paper ***i*** and ***j*** are similar (both about AI)
- fine-grained → paper ***i*** and ***j*** are different (NLP ≠ CV)

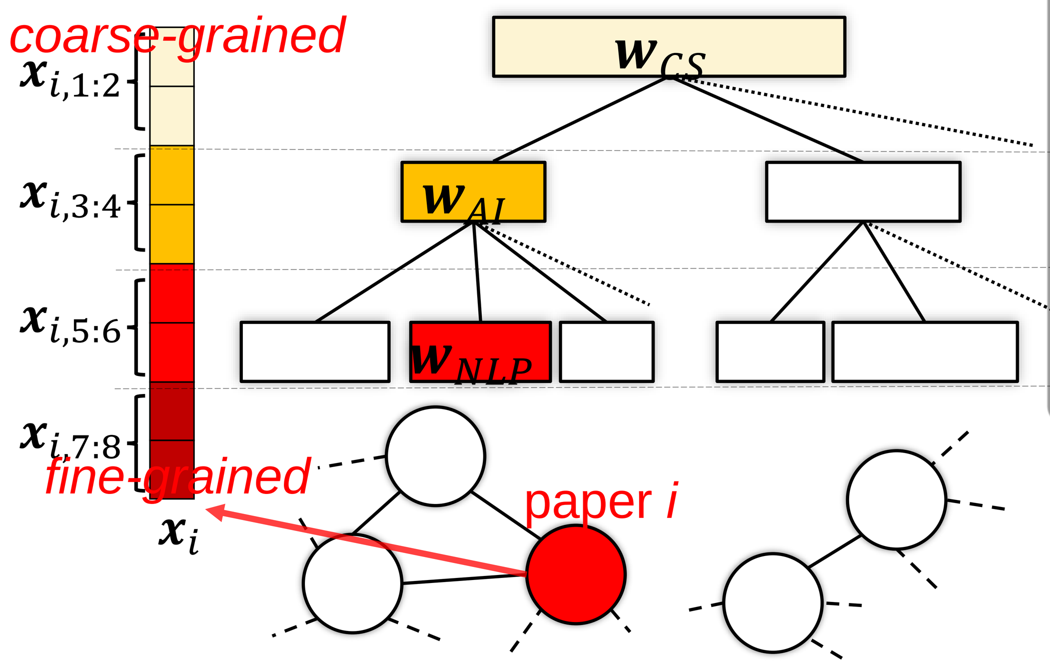
2. Better Interpretability

3 (Method) A Generative Model

Not all networks come with a labeled taxonomy. To learn the taxonomy unsupervisedly, imagine there's *a tree of height* L, where each non-leaf node has an infinite number of child nodes.



A tree with infinite width ($L = 3$).



A toy example ($L = 3, d = 8$).

Each vertex i is associated with a path $\boldsymbol{c}_i$ of length L. We define $p(\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_N)$ as a nested Chinese restaurant process[Blei03], i.e., $\boldsymbol{c}_n \mid \boldsymbol{c}_{1:(n-1)} \sim \mathrm{nCRP}\big(\gamma, \boldsymbol{c}_{1:(n-1)}\big)$. The subtree formed by $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N$ is the hierarchical taxonomy we aim to learn.

Let $\boldsymbol{x}_i \in \boldsymbol{R}^d$ be the representation of vertex $i \in V$. Each node in the infinite-sized tree represents a cluster. Let the representation of cluster t be $\boldsymbol{w}_t \in \boldsymbol{R}^{\Delta d}$, where $\Delta d = \left\lfloor \frac{d}{L+1} \right\rfloor$. The prior over $\boldsymbol{w}_t$ is $\boldsymbol{w}_t \sim \mathrm{Normal}(0, \sigma_w^2 \boldsymbol{I})$ (we use $\sigma_w \to \infty$).

Vertex representation $\boldsymbol{x}_i$ is split into L+1 parts (each of Δd, except the last one). The first L parts (i.e., $\boldsymbol{x}_{i,1:L\Delta d}$) follows $\boldsymbol{x}_{i,1:L\Delta d} \sim \mathrm{Normal}\big(\boldsymbol{w}_{\boldsymbol{c}_i}, \sigma_x^2 \boldsymbol{I}\big)$, where $\boldsymbol{w}_{\boldsymbol{c}_i}$ is the result of concatenating all the $\boldsymbol{w}_t$ visited by path $\boldsymbol{c}_i$. The last part, however, just follows $\boldsymbol{x}_{i,(L\Delta d+1):d} \sim \mathrm{Normal}(\boldsymbol{0}, +\infty \boldsymbol{I})$ (for capturing features that are unique to vertex i itself).

Let $r_{uv} = 1$ if vertex u and v are linked. And we sample $r_{uv} = 0$ via negative sampling. (We can additionally add more $r_{uv} = 1$, by leveraging 2rd-order proximity or random walks). Then,

$$r_{uv} \sim \mathrm{Bernoulli}\left(e^{-\frac{\|x_u - x_v\|^2}{l^2}}\right).$$

4 (Method) How to Optimize

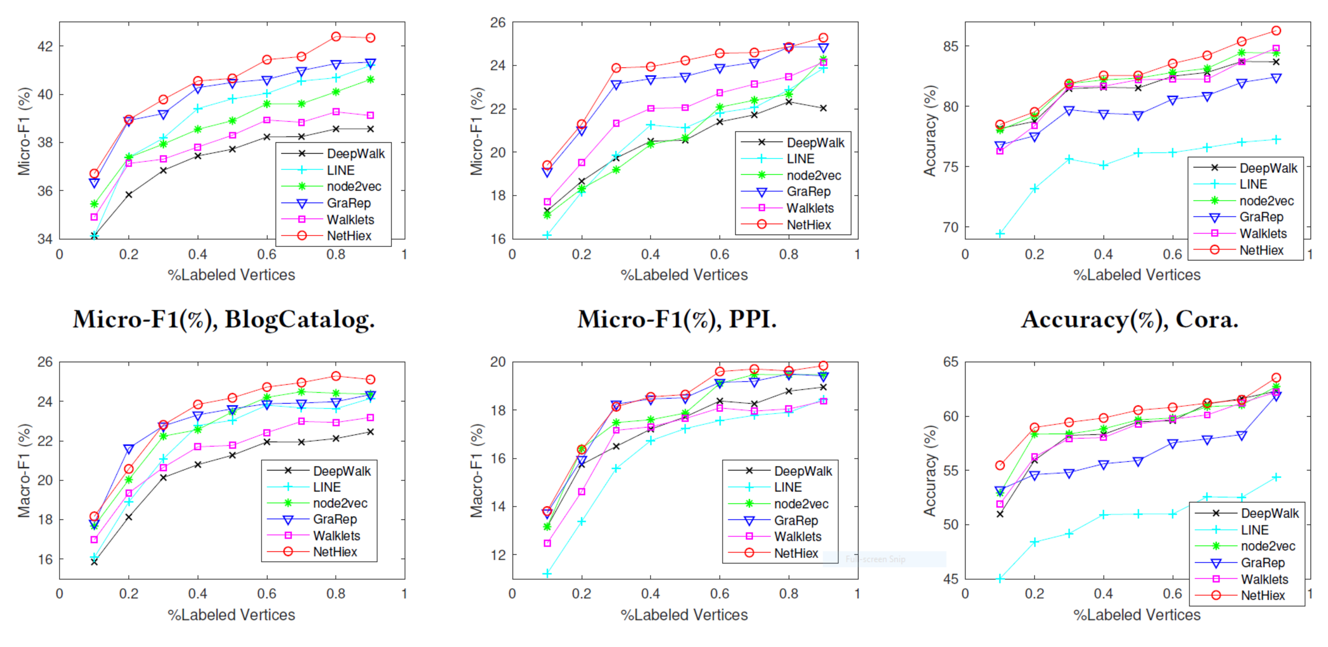
EM Algorithm + Truncated Tree[Wang-Blei-09]

To find $\boldsymbol{x}_i$ and $\boldsymbol{w}_t$ that maximize $\log \sum_C p(X, W, C)$, where $C = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_n\}$ and $p(X, W, C) =$

$$\prod_t p(\boldsymbol{w}_t) \cdot \prod_i p(\boldsymbol{c}_i \mid \boldsymbol{c}_{1:(i-1)}) p(\boldsymbol{x}_i \mid \boldsymbol{w}_{\boldsymbol{c}_i}) \cdot \prod_{uv} p(r_{uv} \mid \boldsymbol{x}_u, \boldsymbol{x}_v).$$

5 (Experiments) Performance

Node Classification



Micro-F1(%), BlogCatalog. Micro-F1(%), PPI. Accuracy(%), Cora.

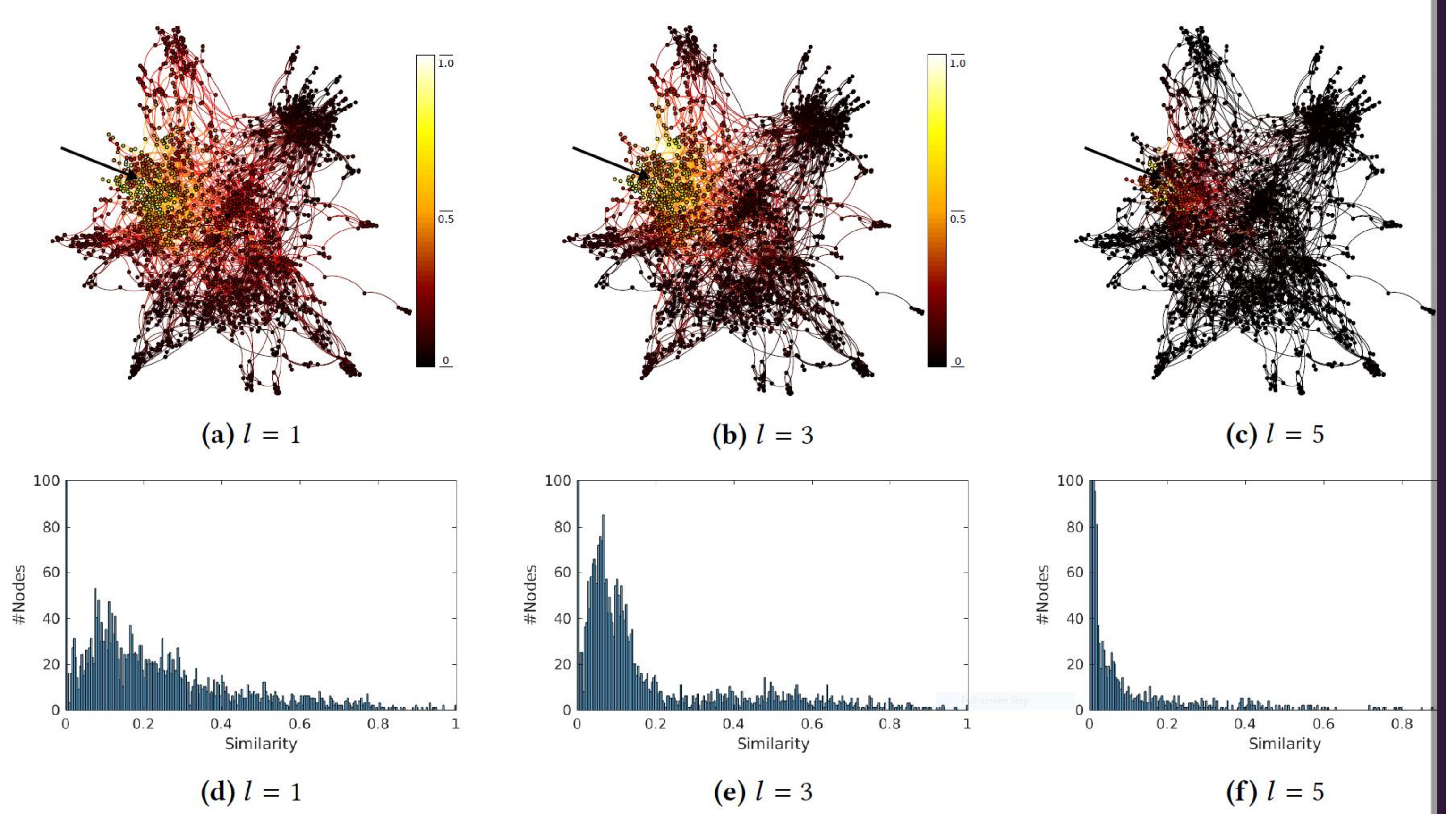
Macro-F1(%), BlogCatalog. Macro-F1(%), PPI. Accuracy(%), Citeseer.

Link Prediction

			Baselines					This Work
Metric	Network	%Missing Links	DeepWalk	LINE	node2vec	GraRep	Walklets	NetHiex
AUC(%)	Citeseer	50%	77.00	77.25	77.58	74.11	74.57	**77.78**
		40%	79.76	80.36	80.04	76.02	78.09	**80.44**
		30%	82.12	82.41	83.03	81.55	80.80	**83.86**
		20%	82.97	84.00	83.02	85.81	83.86	**87.19**
		10%	86.59	88.44	86.74	87.15	86.16	**88.97**
	PPI	50%	74.60	73.23	75.13	76.81	74.55	**76.85**
		40%	75.00	74.34	75.92	77.73	74.19	**78.07**
		30%	75.49	75.13	76.02	77.80	76.37	**77.98**
		20%	76.73	75.35	77.04	78.51	77.89	**78.55**
		10%	77.30	75.69	77.69	78.96	78.89	78.96
	Cora	50%	74.50	73.84	75.16	75.85	71.05	**80.86**
		40%	80.48	78.81	80.61	82.93	76.75	**87.62**
		30%	81.59	81.09	82.37	85.94	78.86	**88.21**
		20%	84.28	82.11	83.72	89.42	81.03	**90.59**
		10%	84.22	83.75	85.03	90.29	81.65	**90.55**

6 (Visualization) Multiple Levels of Granularity

The inter-vertex similarity between a vertex and all the other vertices, in terms of the 1st, 3rd, and 5th parts of the learned vertex representations (Cora, $L = 5$).



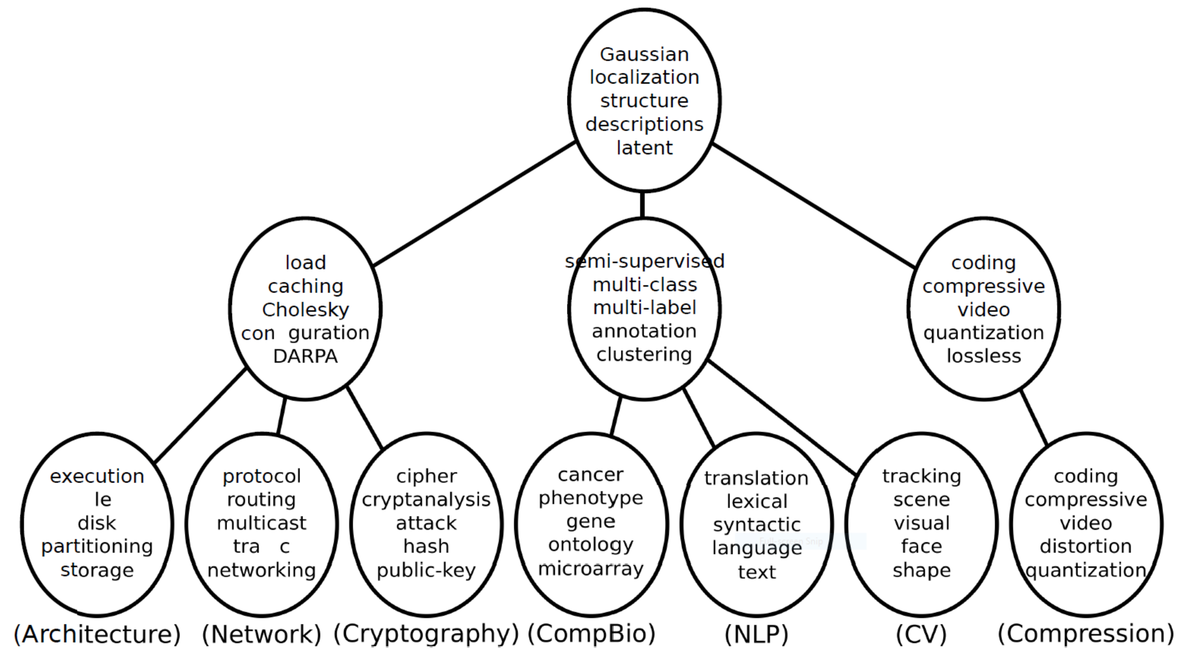
(a) $l = 1$ (b) $l = 3$ (c) $l = 5$

(d) $l = 1$ (e) $l = 3$ (f) $l = 5$

The different components of the vertex representations indeed reflect the different levels of granularity.

7 (Vis.) Finding Out the Taxonomy

We can uncover the hierarchical taxonomy, unsupervisedly, from a word co-occurrence network.



(The word co-occurrence network is constructed from CS paper titles. The words with a low TF-IDF score are removed.)