Python for NLP, Graduate, 2018 Fall

Author2Vec: Learning Author Representations by Combining Content and Link Information

Ganguly, Soumyajit, et al. 2016. (Microsoft, Hyderabad, India)

Byeongki Jeong

Department of industrial engineering @Konkuk university

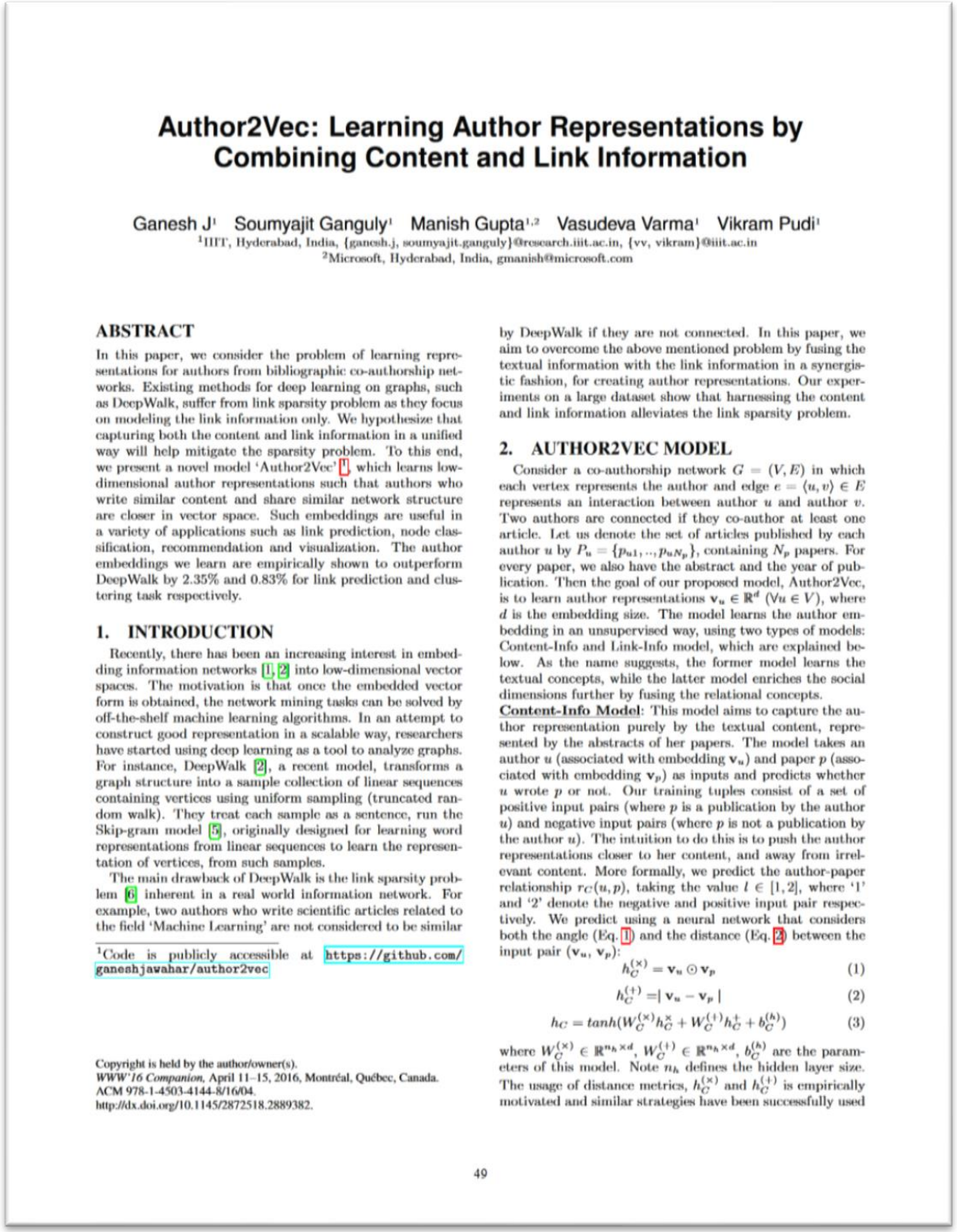
jbk958@gmail.com

http://byeongkijeong.github.io/

Contents

- Introduction
- Author2Vec model
- Experiments
- Conclusions

https://dl.acm.org/citation.cfm?id=2889382
https://github.com/ganeshjawahar/author2vec



Author2Vec: Learning Author Representations by Combining Content and Link Information

Ganesh J[1] Soumyajit Ganguly[1] Manish Gupta[1,2] Vasudeva Varma[1] Vikram Pudi[1]
[1]IIIT, Hyderabad, India, {ganesh.j, soumyajit.ganguly}@research.iiit.ac.in, {vv, vikram}@iiit.ac.in
[2]Microsoft, Hyderabad, India, gmanish@microsoft.com

ABSTRACT

In this paper, we consider the problem of learning representations for authors from bibliographic co-authorship networks. Existing methods for deep learning on graphs, such as DeepWalk, suffer from link sparsity problem as they focus on modeling the link information only. We hypothesize that capturing both the content and link information in a unified way will help mitigate the sparsity problem. To this end, we present a novel model 'Author2Vec' [1], which learns low-dimensional author representations such that authors who write similar content and share similar network structure are closer in vector space. Such embeddings are useful in a variety of applications such as link prediction, node classification, recommendation and visualization. The author embeddings we learn are empirically shown to outperform DeepWalk by 2.35% and 0.83% for link prediction and clustering task respectively.

1. INTRODUCTION

Recently, there has been an increasing interest in embedding information networks [1, 2] into low-dimensional vector spaces. The motivation is that once the embedded vector form is obtained, the network mining tasks can be solved by off-the-shelf machine learning algorithms. In an attempt to construct good representation in a scalable way, researchers have started using deep learning as a tool to analyze graphs. For instance, DeepWalk [2], a recent model, transforms a graph structure into a sample collection of linear sequences containing vertices using uniform sampling (truncated random walk). They treat each sample as a sentence, run the Skip-gram model [5], originally designed for learning word representations from linear sequences to learn the representation of vertices, from such samples.

The main drawback of DeepWalk is the link sparsity problem [6] inherent in a real world information network. For example, two authors who write scientific articles related to the field 'Machine Learning' are not considered to be similar by DeepWalk if they are not connected. In this paper, we aim to overcome the above mentioned problem by fusing the textual information with the link information in a synergistic fashion, for creating author representations. Our experiments on a large dataset show that harnessing the content and link information alleviates the link sparsity problem.

2. AUTHOR2VEC MODEL

Consider a co-authorship network $G = (V, E)$ in which each vertex represents the author and edge $e = \langle u, v\rangle \in E$ represents an interaction between author u and author v. Two authors are connected if they co-author at least one article. Let us denote the set of articles published by each author u by $P_u = \{p_{u1}, .., p_{uN_p}\}$, containing N_p papers. For every paper, we also have the abstract and the year of publication. Then the goal of our proposed model, Author2Vec, is to learn author representations $\mathbf{v}_u \in \mathbb{R}^d$ $(\forall u \in V)$, where d is the embedding size. The model learns the author embedding in an unsupervised way, using two types of models: Content-Info and Link-Info model, which are explained below. As the name suggests, the former model learns the textual concepts, while the latter model enriches the social dimensions further by fusing the relational concepts.

Content-Info Model: This model aims to capture the author representation purely by the textual content, represented by the abstracts of her papers. The model takes an author u (associated with embedding $\mathbf{v}_u$) and paper p (associated with embedding $\mathbf{v}_p$) as inputs and predicts whether u wrote p or not. Our training tuples consist of a set of positive input pairs (where p is a publication by the author u) and negative input pairs (where p is not a publication by the author u). The intuition to do this is to push the author representations closer to her content, and away from irrelevant content. More formally, we predict the author-paper relationship $r_C(u, p)$, taking the value $l \in [1, 2]$, where '1' and '2' denote the negative and positive input pair respectively. We predict using a neural network that considers both the angle (Eq. 1) and the distance (Eq. 2) between the input pair $(\mathbf{v}_u, \mathbf{v}_p)$:

$$h_C^{(\times)} = \mathbf{v}_u \odot \mathbf{v}_p \quad (1)$$

$$h_C^{(+)} = \mid \mathbf{v}_u - \mathbf{v}_p \mid \quad (2)$$

$$h_C = tanh(W_C^{(\times)} h_C^{\times} + W_C^{(+)} h_C^{+} + b_C^{(h)}) \quad (3)$$

where $W_C^{(\times)} \in \mathbb{R}^{n_h \times d}$, $W_C^{(+)} \in \mathbb{R}^{n_h \times d}$, $b_C^{(h)}$ are the parameters of this model. Note n_h defines the hidden layer size. The usage of distance metrics, $h_C^{(\times)}$ and $h_C^{(+)}$ is empirically motivated and similar strategies have been successfully used

[1] Code is publicly accessible at `https://github.com/ganeshjawahar/author2vec`

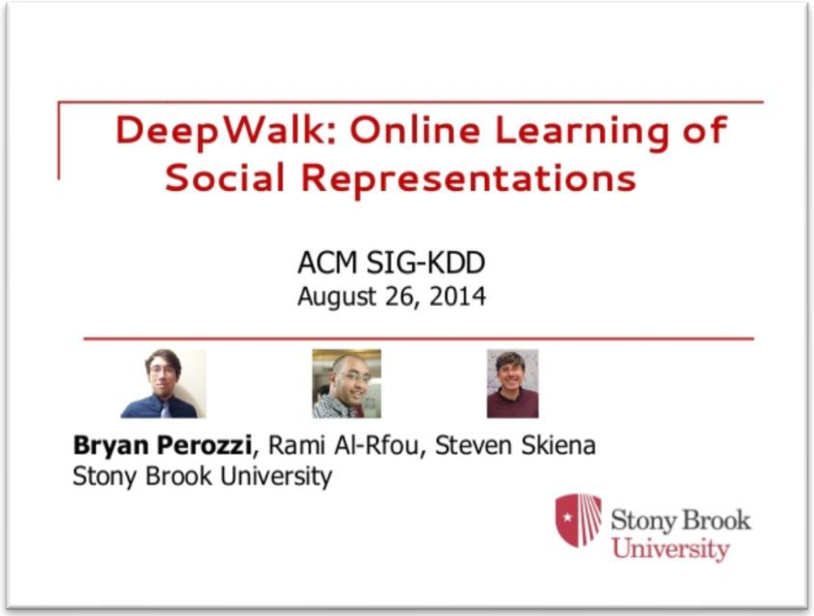
WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2889382.

세줄요약:

1) 기존 Graph representation은 Link information만 사용
2) Link information은 Sparse하기 때문에 한계가 있음
3) Contents를 같이 이용한 Author representation 제안

Introduction

- 네트워크를 저차원 벡터로 Embedding하려는 목적으로 다양한 방법이 시도됨
  - Embedding된 벡터를 이용하면 기존 머신러닝 알고리즘으로 네트워크 분석 가능
  - 특히 최근에는 Deep neural network를 Embedding에 많이 사용함
- 특히, DeepWalk는 Graph embedding의 SOTA임(2016년 기준)
  - Graph상에서 짧은 Random walk를 수행하여 Vertex의 Sequence 생성
  - 만들어진 Sequence를 문장처럼 인식하여 Skip-gram으로 학습

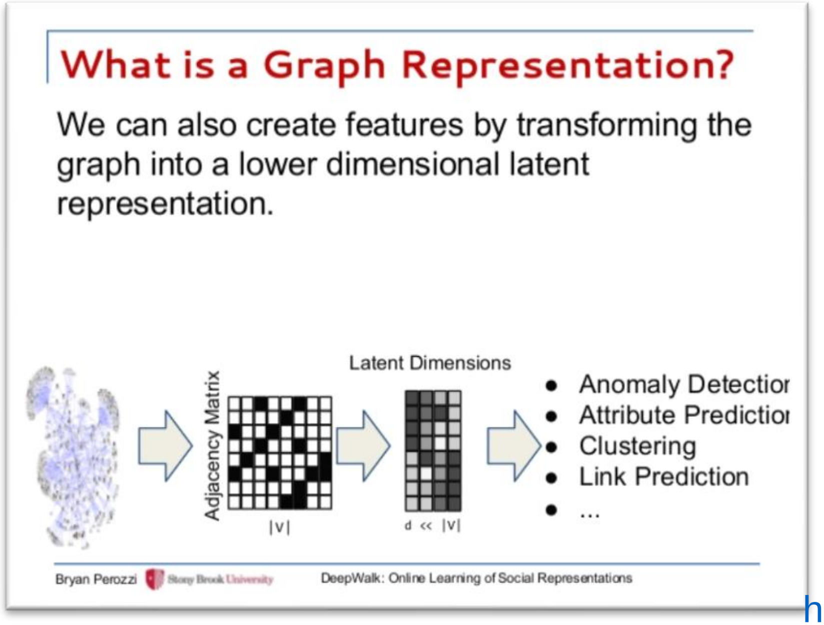


DeepWalk: Online Learning of Social Representations

ACM SIG-KDD
August 26, 2014

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

Stony Brook University

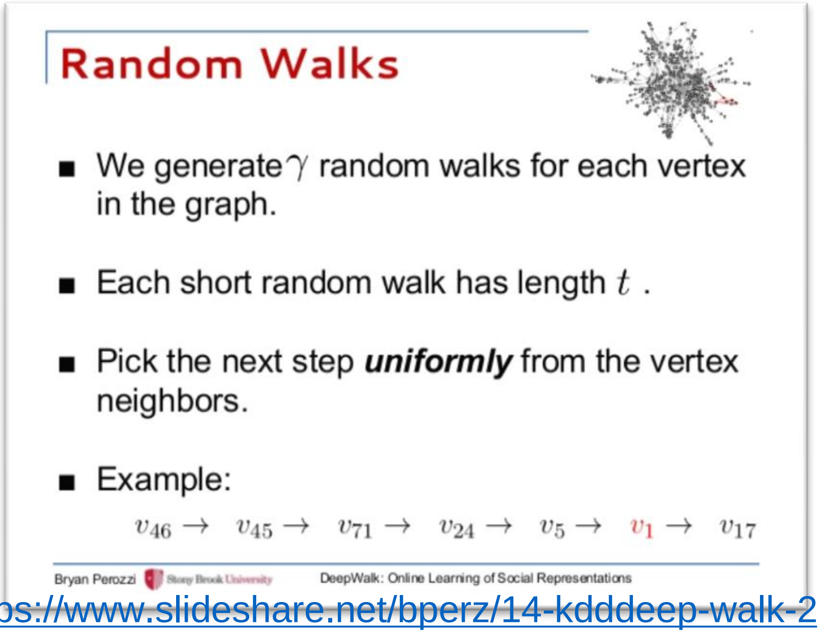


What is a Graph Representation?

We can also create features by transforming the graph into a lower dimensional latent representation.

- Anomaly Detectior
- Attribute Predictior
- Clustering
- Link Prediction
- ...

Bryan Perozzi Stony Brook University DeepWalk: Online Learning of Social Representations



Random Walks

- We generate γ random walks for each vertex in the graph.
- Each short random walk has length t .
- Pick the next step ***uniformly*** from the vertex neighbors.
- Example:

$$v_{46} \rightarrow v_{45} \rightarrow v_{71} \rightarrow v_{24} \rightarrow v_5 \rightarrow v_1 \rightarrow v_{17}$$

Bryan Perozzi Stony Brook University DeepWalk: Online Learning of Social Representations

https://www.slideshare.net/bperz/14-kdddeep-walk-2

Link information은 **Sparse**하기 때문에,
직접 연결된 적이 없으면
비슷한 Node라도 유사하게 Embedding 되지 않음

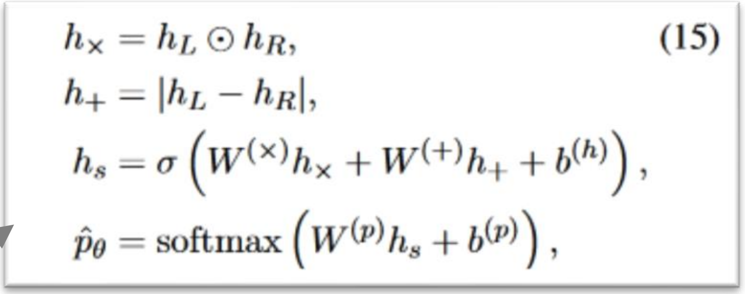
비슷한 연구를 하고 있는 Authors여도, 공저자였던 적이 없으면 Random walk sequence가 만들어질 리가 없으므로…

Author2Vec

- Link sparsity 극복을 위해 Textual information과 Link information을 함께 사용한 Author2Vec을 제안
  - $G = (V, E)$
    - $V = Authors$
    - $e =< u, v >\in E\text{: } Co - authorship$
  - $P_u = \left\{p_{u1}, \ldots, p_{uN_p}\right\}\text{: } Set\ of\ articles\ published\ by\ author\ u$
    - 각각의 Article은 초록, 출판년도 보유
  - Author2Vec의 목표는 $Author\ representations, \boldsymbol{v}_u \in \mathbb{R}^d (\forall u \in V)$를 학습하는 것
    - $d = Embedding\ size$
  - Content-Info model과 Link-Info model, 두가지 모형을 학습해서 Author2Vec에 사용
    - Content-Info model: Textual concepts 학습
    - Link-Info model: Social dimensions 학습

Author2Vec: Content-Info model

- Textual information($\boldsymbol{v}_p$)을 이용하여 Author representation ($\boldsymbol{v}_u$) 을 만드는 모형
  - Input: ($\boldsymbol{v}_u$, $\boldsymbol{v}_p$), positive pair and negative pair
  - Output: $r_C(u,p) = l, l \in [1,2]$, 2: positive and 1: negative
  - Model: $h_C = tanh(W_C^{(\times)} h_C^{(\times)} + W_C^{(+)} h_C^{(+)} + b_C^{(h)})$
    - $W_C^{(\times)} \in \mathbb{R}^{n_h \times d}, W_C^{(+)} \in \mathbb{R}^{n_h \times d}$
      - $n_h = Hidden\ layer\ size,\ d = Embedding\ size$
    - $h_C^{(\times)} = \boldsymbol{v}_u \odot \boldsymbol{v}_p$: Angle of two vectors
    - $h_C^{(+)} = |\boldsymbol{v}_u - \boldsymbol{v}_p|$: Distance of two vectors
      - Tai et al. "Improved semantic representations from tree-structured long short-term memory networks." arXiv preprint arXiv:1503.00075 (2015).



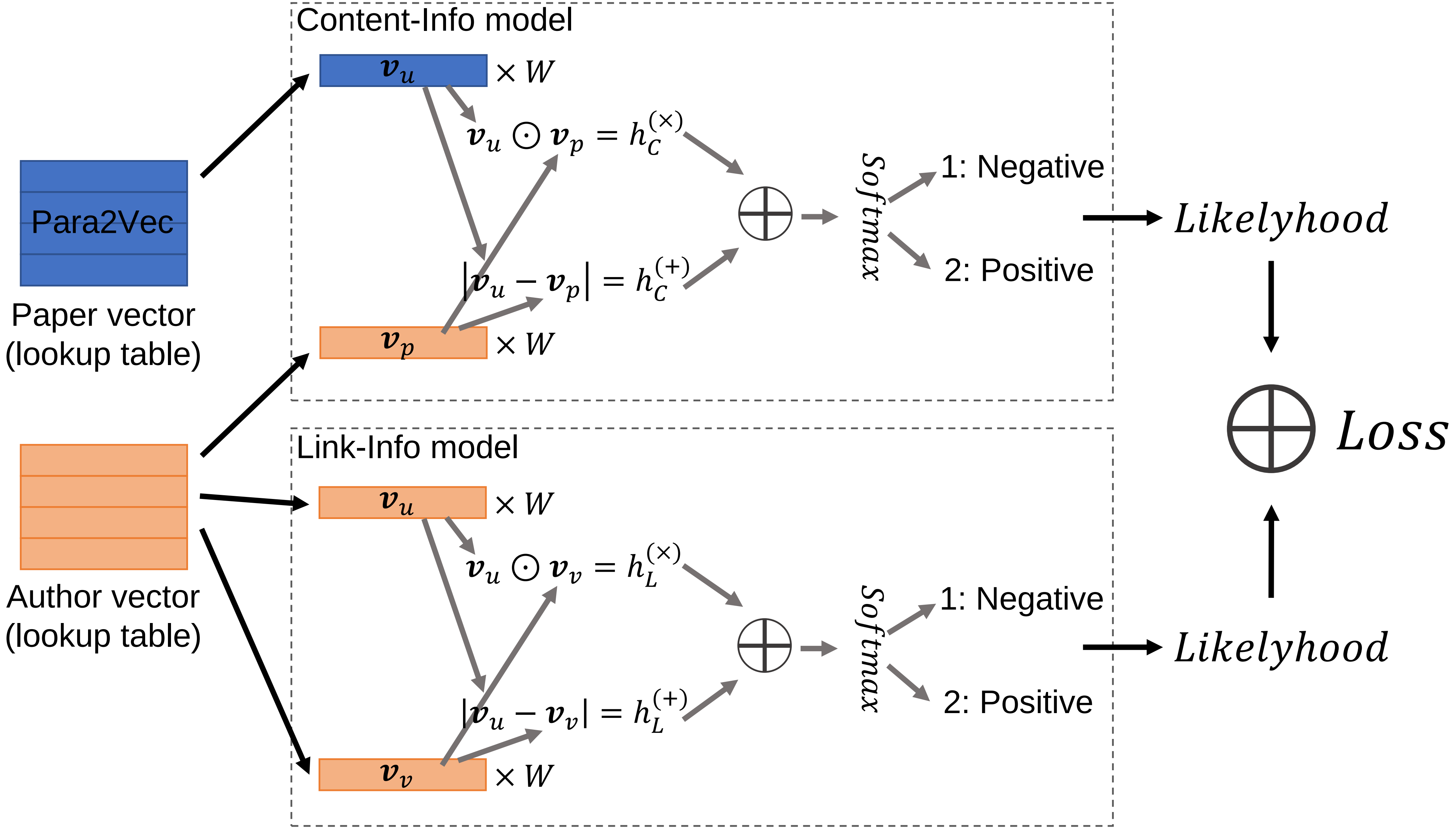
$$h_\times = h_L \odot h_R, \qquad (15)$$
$$h_+ = |h_L - h_R|,$$
$$h_s = \sigma\left(W^{(\times)}h_\times + W^{(+)}h_+ + b^{(h)}\right),$$
$$\hat{p}_\theta = \text{softmax}\left(W^{(p)}h_s + b^{(p)}\right),$$

  - Loss function: $\mathcal{L}_C = \mathrm{P}[r_C(u,p) = l] = softmax(U_C h_C + b_C^{(p)})$
    - $U_C \in \mathbb{R}^{n_h \times d}$
  - Paper represeantation($\boldsymbol{v}_p$)는 Paragraph2Vec으로 Pre-initialized 된 값을 사용

Author2Vec: Link-Info model

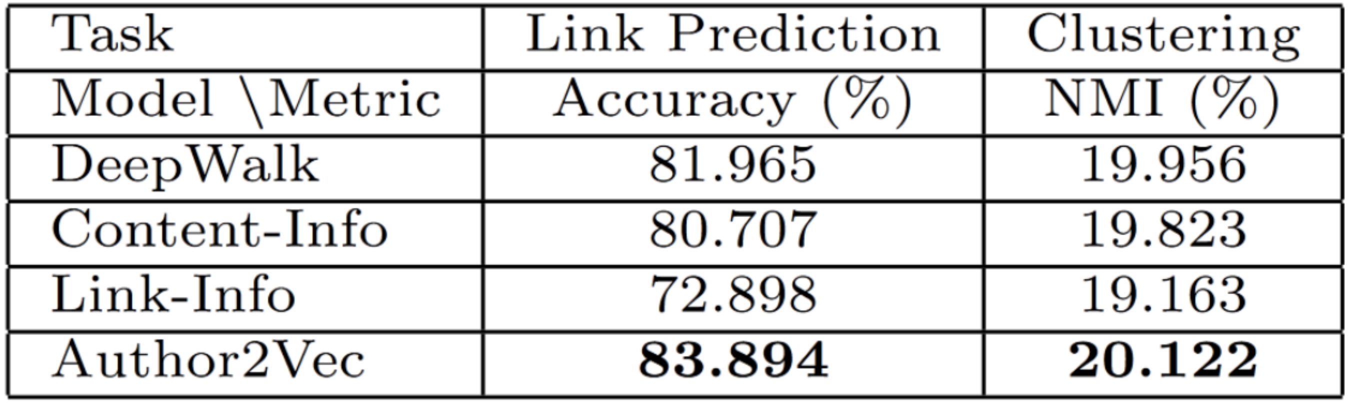
- 공저자였던 두 Author vector를 유사하게 조정하는 모형
    - Input: $(\boldsymbol{v}_u, \boldsymbol{v}_v)$, positive pair and negative pair
    - Output: $r_L(u, v) = l, l \in [1,2]$, 2: positive and 1: negative
    - Model: $h_C = tanh(W_L^{(\times)} h_L^{(\times)} + W_L^{(+)} h_L^{(+)} + b_L^{(h)})$
        - $W_L^{(\times)} \in \mathbb{R}^{n_h \times d}, W_L^{(+)} \in \mathbb{R}^{n_h \times d}$
            - $n_h = Hidden\ layer\ size,\ d = Embedding\ size$
        - $h_L^{(\times)} = \boldsymbol{v}_u \odot \boldsymbol{v}_v$: Angle of two vectors
        - $h_L^{(+)} = |\boldsymbol{v}_u - \boldsymbol{v}_v|$: Distance of two vectors
    - Loss function: $\mathcal{L}_L = \mathrm{P}[r_L(u, v) = l] = softmax(U_L h_L + b_L^{(p)})$
        - $U_L \in \mathbb{R}^{n_h \times d}$
- Content-Info model과 Link-Info model을 함께 이용하여 Author embedding을 학습함
    - 두 모형이 Author embedding을 공유하는 형태로 학습이 진행 (Loss function: $\mathcal{L} = \mathcal{L}_C + \mathcal{L}_L$)

Author2Vec: Graphical representation



Experiments

- DeepWalk와 성능을 비교
  - Citation network dataset (Chakraborty et al. 2013)을 이용
    - 1990~2010기간동안 CS 논문 Dataset
    - 각 논문은 저자, 초록, 발행년도를 가지며 24개 CS분야 중 하나에 속함
  - Link prediction(Logistic regression), Clustering(K-means, k=24) 성능을 측정하여 비교
    - Link prediction은 정확도로 측정, Clustering은 NMI로 측정
      - $NMI(Y,C) = \frac{2 \times I(Y;C)}{H(Y)+H(C)}$
  - 개별 모형의 성능은 DeepWalk보다 떨어졌지만, Author2Vec은 DeepWalk보다 우위를 보임



Task	Link Prediction	Clustering
Model \Metric	Accuracy (%)	NMI (%)
DeepWalk	81.965	19.956
Content-Info	80.707	19.823
Link-Info	72.898	19.163
Author2Vec	**83.894**	**20.122**

Conclusions

- Author2Vec은 Link정보와 Content정보를 함께 사용하여 기존 DeepWalk보다 성능상 우위를 보임

- 발표자 의견
  - DeepWalk의 접근법은 간단하면서 파워풀함
  - Author2Vec은 시도 자체로 의의가 있지만, Graph embedding에 적용하기에는 한계가 있음
    - Textual information 이 없는 Graph에는 적용 불가
    - 그래서 목적을 Author embedding으로 한정지은 것 일지도 모르겠음
  - 결정적으로 DeepWalk에 비해 복잡한 만큼 성능차이가 큰 것 같지 않음
    - DeepWalk를 더 공부해보고 싶음