CARL: Content-Aware Representation Learning for Heterogeneous Networks

Chuxu Zhang
University of Notre Dame
czhang11@nd.edu

Ananthram Swami
Army Research Laboratory
ananthram.swami.civ@mail.mil

Nitesh V. Chawla
University of Notre Dame
nchawla@nd.edu

ABSTRACT

Heterogeneous networks not only present a challenge of heterogeneity in the types of nodes and relations, but also the attributes and content associated with the nodes. While recent works have looked at representation learning on homogeneous and heterogeneous networks, there is no work that has collectively addressed the following challenges: (a) the heterogeneous structural information of the network consisting of multiple types of nodes and relations; (b) the unstructured semantic content (e.g., text) associated with nodes; and (c) online updates due to incoming new nodes in growing network. We address these challenges by developing a Content-Aware Representation Learning model (CARL). CARL performs joint optimization of heterogeneous SkipGram and deep semantic encoding for capturing both heterogeneous structural closeness and unstructured semantic relations among all nodes, as function of node content, that exist in the network. Furthermore, an additional online update module is proposed for efficiently learning representations of incoming nodes. Extensive experiments demonstrate that CARL outperforms state-of-the-art baselines in various heterogeneous network mining tasks, such as link prediction, document retrieval, node recommendation and relevance search. We also demonstrate the effectiveness of the CARL’s online update module through a category visualization study.

KEYWORDS

Heterogeneous Information Networks, Representation Learning, Network Embedding

1 INTRODUCTION

Heterogeneous information networks (HetNets) [16, 17], e.g., academic networks, encode rich information through multi-typed nodes, relationships, and attributes or content associated with nodes. For example, the academic networks can represent human-human relationship (authors), human-object relationship (author-paper or author-venue or author-organization), and object-object relationship (paper-paper, paper-venue, paper-organization). The nodes in this case (human and object) can carry attributes or semantic content (such as paper abstract). Given the multi-typed nodes, relationships, and content at the nodes, feature engineering has presented a unique challenge for network mining tasks such as relation mining [15, 23], relevance search [7, 16], personalized recommendation [9, 14, 27]. The typical feature engineering activity responds to the requirements of the network mining task for an application, requiring both a domain understanding and large exploratory search space for possible features. Not only this is expensive, but it also may not result in optimal performance.

To that end, we ask the question: Can we generalize the feature engineering activity through representation learning on HetNets that addresses the complexity of multi-typed data in HetNets? With the advent of deep learning, significant effort has been devoted to network representation learning in the last few years, starting with a focus on homogeneous networks [5, 12, 20] and more recently on HetNets [4]. The underlying theme of the models developed in these works is to automate the discovery of useful node latent features that can be further utilized in various network mining problems such as link prediction and node recommendation. However, these methods are limited in truly addressing the challenges of HetNets:

- **(C1)** HetNets include multiple types of nodes and relations. For example, in Figure 1(a), an academic network involves three types of nodes, i.e., author, paper and venue, which are connected by three types of relations, i.e., author-write-paper, paper-cite-paper and paper-publish-venue. Most of the previous models (e.g., Deepwalk and node2vec) employ homogeneous language models which make application to HetNets difficult. Thus challenge 1 is: how to extend homogeneous language model to heterogeneous network representation learning for maintaining structural closeness among multiple types of nodes and relations? We build on our prior work metapath2vec [4] for this.

- **(C2)** HetNets include both structural content (e.g., node type and relation connection) and unstructured semantic content (e.g., text). For example, in Figure 1(a), paper in academic network connects to author & venue and contains semantic text. The current models purely depend on structural content yet can not leverage unstructured content to infer semantic relations that are far away in network. To be more specific (as we will show in Section 4.5), given query author “Jure Leskovec”, conventional techniques (e.g., node2vec and metapath2vec) tend to}
return authors who collaborated with “Jure” due to structural relations bridged by paper or return authors who are different from “Jure” in specific research interests due to structural relations bridged by venue. Thus challenge 2 is: how to effectively incorporate unstructured content of nodes into a representation learning framework for capturing both structural closeness and unstructured semantic relations among all nodes?

• (C3) HetNets can grow with time. Current models are not able to handle this due to lack of an update strategy and it is impractical to re-run the model for each new node. For example, in Figure 1(b), new author A5 co-authors with A1 and A4 on new paper P5 after a given time split T. Thus challenge 3 is: how to efficiently learn representations of new nodes in a growing network?

Our proposed method CARL, a content-aware representation learning model for HetNets, addresses these challenges. Specifically, first, we develop a heterogeneous SkipGram model to maintain structural closeness among multiple types of nodes and relations. Next, we design two effective ways based on deep semantic encoding to incorporate unstructured content (i.e., text) of some types of nodes into heterogeneous SkipGram for capturing semantic relations. The negative sampling technique and the walk sampling based strategy are utilized to optimize and train the proposed models. Finally, we develop an online update module to efficiently learn representation of each new node by using its relations with existing nodes and the learned node representations.

To summarize, the main contributions of our work are:

• We formalize the problem of content-aware representation learning in HetNets and develop a model, i.e., CARL, to solve the problem. CARL performs joint optimization of heterogeneous SkipGram and deep semantic encoding.
• We design the corresponding optimization strategy and training algorithm to effectively learn node representations. The output representations are further utilized in various HetNet mining tasks, such as link prediction, document retrieval, node recommendation and relevance search, which demonstrate the superior performance of CARL over state-of-the-art baselines.
• We propose an update module in CARL to handle growing networks and conduct the category visualization study to show the effectiveness of this module.

2 PROBLEM DEFINITION

We first introduce the concepts of HetNets and random & meta-path walks, then formally define the problem of content-aware representation learning in HetNets.

Definition 2.1. (Random Walk) A random walk [5] is defined as a node sequence \( S_{v_0} = \{v_0, v_1, v_2, \ldots, v_{L-1}\} \) wherein the \( i \)-th node \( v_{i-1} \) in the walk is randomly selected from the neighbors of its predecessor \( v_{i-2} \).

Definition 2.2. (Meta-path Walk) A meta-path walk [4] in HetNet is defined as a random walk guided by a specific meta-path scheme with the form of \( P \equiv o_1 \rightarrow o_2 \rightarrow \cdots \rightarrow o_m \), where \( o_1 \in O_V, r_1 \in R_E \) and \( r = r_1 \ast r_2 \cdots \ast r_{m-1} \) represents a compositional relation between relation types \( r_1 \) and \( r_m \). Each meta-path walk recursively samples a specific \( P \) until it meets the given length.

Figure 2(b) shows examples of random walk and “APVPA” meta-path walk in the academic network.

Definition 2.4. (Content-Aware Representation Learning in Heterogeneous Networks) Given a HetNet with both structural and unstructured content at each node, the task is to design a model to learn a \( d \)-dimensional feature representations \( \theta \in \mathbb{R}^{V \times d} (d \ll |V|) \), which can encode both structural closeness and unstructured semantic relations. Furthermore, the model is able to efficiently infer representation \( \Theta_v' \) of each new node \( v' \) by using the learned representations \( \theta \).

For example, in the network of Figure 1, author and venue nodes contain structural content, i.e., node id, node type as well as link relations with others, and paper node contains both structural content and unstructured semantic content, e.g., abstract text. The output \( \theta \) denotes representations of all existing nodes via the same latent space, which can be further utilized in various HetNet mining tasks. Besides, the learned representation \( \Theta_v' \) of each new node \( v' \) can benefit different tasks for \( v' \) such as category assignment.

3 CARL FRAMEWORK

We present the framework of content-aware representation learning which will address the three challenges described in Section 1.

3.1 Heterogeneous Network Embedding (C1)

Inspired by word2vec [10] for learning distributed representation of words in corpus, Deepwalk [12] and node2vec [5] leverage SkipGram and random walks to learn node representations. However, we argued that those techniques focus on homogeneous networks and proposed metapath2vec [4] for HetNets by feeding meta-path walks to SkipGram. Similar to metapath2vec, we formulate the heterogeneous network representation learning as heterogeneous SkipGram (HSG) to address challenge C1. Specifically, given a HetNet \( G = (V, E, O_V, R_E) \), the objective is to maximize the likelihood of each type of context node given the input node \( v \):

\[
O_1 = \arg\max_{\theta} \prod_{v \in V} \prod_{e \in E} \prod_{v_i \in N_E(v)} p(v_i|v;\theta) \tag{1}
\]

where \( \theta \) contains the representations of all nodes and \( N_E(v) \) is the set of \( t \)-type context node of \( v \) which can be collected in different ways such as one-hop neighbors or surrounding neighbors in random walks. For example, in Figure 2(b), A5 is structurally close to other authors (e.g., A1 & A4), papers (e.g., P2 & P4) and venues (e.g., \( V_1 \& V_5 \)). Thus objective \( O_1 \) is able to maintain structural closeness among multiple types of nodes and relations in \( G \).
3.2 Incorporating Deep Semantic Encoder (C2)

The objective $o_1$ formulates structural closeness but ignores unstructured semantic relations. To address challenge C2, we design two ways to incorporate unstructured content of some types of nodes into heterogeneous network representation learning.

3.2.1 HSG with Unstructured Semantic Regularization (HSG-SR). One way is to tightly join HSG with the conditional probability of semantic constraint, leading to unstructured semantic regularization onto objective $o_1$. The objective is defined as:

$$o_2 = \arg\max_{\theta, \Phi} \prod_{v \in V} \prod_{c \in N(v)} p(v_c | v; \theta) \prod_{v \in V_S} p(\theta_v | Y_v; \Phi)$$

(2)

where $V_S$ is the set of nodes with unstructured semantic content, $Y_v$ represents unstructured content of node $v$ and $\Phi$ are parameters of a deep semantic encoder that will be described later. The conditional probability $p(v_c | v; \theta)$ is defined as the heterogeneous softmax function:

$$p(v_c | v; \theta) = \frac{e^{\theta_{vc} \theta_v}}{\sum_{v' \in V_c} e^{\theta_{vc} \theta_{v'}}}$$

where $V_c$ is the set of neighbors of node $v$. Besides, we model the conditional probability $p(\theta_v | Y_v; \Phi)$ as Gaussian prior: $p(\theta_v | Y_v; \Phi) = N(\theta_v | E_v, \sigma^2 I)$, where $E_v$ denotes $v$’s semantic representation encoded by deep learning architecture $f$: $E_v = f(Y_v)$. For example, in the network of Figure 2, $V_S$ in $o_2$ is the set of papers and the formulation involves four kinds of representations, i.e., author representations, venue representations, paper representations and paper semantic representations. Notice that, there are two kinds of paper representations and we will use paper semantic representation for evaluation in Section 4.

3.2.2 Unstructured Semantic Enhanced HSG (SE-HSG). Another way is to concatenate the output of deep semantic encoder with the input of HSG, leading to unstructured semantic enhancement onto objective $o_1$. The objective is defined as:

$$o_3 = \arg\max_{\theta, \Phi} \prod_{v \in V} \prod_{c \in N(v)} p(v_c | v; \theta, Y; \Phi)$$

(3)

where $Y$ is the set of all unstructured semantic content of $V_S$. The conditional probability $p(v_c | v; \theta, Y; \Phi)$ is defined as the semantic enhanced heterogeneous softmax function:

$$p(v_c | v; \theta, Y; \Phi) = \frac{e^{\theta_{vc} \theta_v + \theta_{yc} y_c}}{\sum_{v' \in V_c} e^{\theta_{vc} \theta_{v'} + \theta_{yc} y_c}}$$

where $\theta$ denotes the enhanced representations.

That is, $\Theta_v = E_v = f(Y_v)$ for $v \in V_S$ otherwise $\Theta_v = \theta_v$. For example, in the network of Figure 2, $Y$ in $o_3$ is text content of all papers and the formulation involves three kinds of representations, i.e., author representations, venue representations and paper semantic representations. Notice that, $\theta$ in $o_3$ only denotes representations of nodes without unstructured content (e.g., author and venue in academic network), which is a bit different from $o_2$.

3.2.3 Unstructured Semantic Content Encoder. Both objectives $o_2$ and $o_3$ involve deep semantic encoding architecture. To encode unstructured content of some types of nodes into fixed length representations $E \in \mathbb{R}^{V \times d}$, we introduce gated recurrent units (GRU), a specific type of recurrent neural network, which has been widely adopted for many applications such as machine translation [3]. Figure 2(a) gives an illustrative example of this encoder for papers in the academic network. To be more specific, each paper’s abstract is represented as a sequence of words: $\{w_1, w_2, \ldots, w_{|Y_2|}\}$, followed by the word embeddings sequence: $\{x_1, x_2, \ldots, x_{|Y_2|}\}$ trained by word2vec [10], where $|Y_2|$ is the maximum length of text. For each step $t$ with the input word embedding $x_t$ and previous hidden state vector $h_t$, the current hidden state vector $h_t$ is updated by $h_t = GRU(x_t, h_{t-1})$, where the GRU module is defined as:

$$z_t = \sigma(A_z x_t + B_z h_{t-1})$$

$$r_t = \sigma(A_r x_t + B_r h_{t-1})$$

$$\tilde{h}_t = \tanh(A_h x_t + B_h (r_t \circ h_{t-1}))$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

(4)

where $\sigma$ is the sigmoid function, $A$ and $B$ are parameter matrices of GRU network (i.e., $\Phi$ in objectives $o_2$ and $o_3$ includes $A$ and $B$), operator $\circ$ denotes element-wise multiplication, $z_t$ and $r_t$ are update gate vector and reset gate vector, respectively. The GRU network encodes word embeddings to deep semantic embeddings $h \in \mathbb{R}^{max \times d}$, which is concatenated with a mean pooling layer to obtain the general semantic representation of paper. All of these steps construct the deep semantic encoder $f$. We have also explored other encoding architectures such as LSTM, bidirectional GRU and attention-based GRU, and obtain similar results. Thus we choose GRU since it has a concise structure and reduce training time.

3.3 Model Optimization and Training

We leverage the negative sampling technique [10] to optimize model and introduce the walk sampling based strategy for model training.
3.3.1 Optimization of HSG-SR. By applying negative sampling to the construction of softmax function, we can approximate the logarithm of $p(v_c|v; \Theta)$ in objective $o_2$ as:

$$
\log \sigma(\Theta_{v_c} \cdot \Theta_v) + \sum_{m=1}^{M} E_{v_m'}-p_{v_m'}(v_c) \log \sigma(-\Theta_{v_m'} \cdot \Theta_v)
$$

(5)

where $M$ is the negative sample size and $P_{v_m'}(v_c)$ is the pre-defined sampling distribution w.r.t. the $t$-type node. In our case, $M$ makes little impact on the performance of proposed models. Thus we set $M = 1$ and obtain the cross entropy loss for optimization:

$$
\log p(v|v_c; \Theta) = \log \sigma(\Theta_{v_c} \cdot \Theta_v) + \log \sigma(-\Theta_{v_m'} \cdot \Theta_v)
$$

(6)

That is, for each context node $v_c$ of $v$, we sample a negative node $v_m'$ according to $P_{v_m'}(v_c)$. Besides, as $p(\Theta_{v}|Y; \Phi) = N(\Theta_{v}|E_v, \sigma^2 I)$, the logarithm of $p(\Theta_{v}|Y; \Phi)$ in objective $o_2$ is equivalent to:

$$
\log p(v_c|Y; \Phi) = \left[\Theta_{v_c} - E_v\right]^T \left[\Theta_v - E_v\right]
$$

where $E_v = f(Y_v)$ and $f$ is the deep semantic encoder. Therefore we rewrite objective $o_2$ as:

$$
o_2 = \sum_{(v, v_c, v_m') \in T_{walk}} \left( \log \sigma(\Theta_{v_c} \cdot \Theta_v) + \log \sigma(-\Theta_{v_m'} \cdot \Theta_v) \right)
$$

$$\quad - \gamma \sum_{v_c \in T_{tri}^{v}} \left[\Theta_{v_c} - f(Y_{v_c})\right]^T \left[\Theta_v - f(Y_v)\right]
$$

(8)

where $\gamma$ is a trade-off factor and $T_{walk}$ denotes the set of triplets $(v, v_c, v_m')$ collected by walk sampling on HetNet, which will be described later. Besides, $T_{tri}^{v}$ is the set of nodes with unstructured content in each triplet of $T_{walk}$ for semantic regularization. That is, $\max(|T_{tri}^{v}|) = 3$.

3.3.2 Optimization of SE-HSG. Similar to HSG-SR, the logarithm of $p(v_c|v; \Theta; Y; \Phi)$ in objective $o_3$ is approximated by:

$$
\log p(v_c|v; \Theta; Y; \Phi) = \log \sigma(\Theta_{v_c} \cdot \Theta_v) + \log \sigma(-\Theta_{v_m'} \cdot \Theta_v)
$$

(9)

where $\Theta_{v_c} = E_{v_c} = f(Y_{v_c})$ for nodes with unstructured content otherwise $\Theta_{v_c} = \Theta_v$. Therefore we rewrite objective $o_3$ as:

$$
o_3 = \sum_{(v, v_c, v_m') \in T_{walk}} \log \sigma(\Theta_{v_c} \cdot \Theta_v) + \log \sigma(-\Theta_{v_m'} \cdot \Theta_v)
$$

(10)

As in HSG-SR, $T_{walk}$ is the set of triplets $(v, v_c, v_m')$ collected from walk sequences on HetNet.

3.3.3 Model Training. Both optimized objectives $o_2$ and $o_3$ are accumulated on set $T_{walk}$. Similar to DeepWalk, node2vec and metapath2vec, we design a walk sampling strategy to generate $T_{walk}$. Specifically, first, we uniformly generate a set of random walks or meta-paths walks $S$ in HetNet. Then, for each node $v$ in $S$, we collect context node $v_c$ which satisfies: $\text{dist}(v, v_c) \leq \tau$. That is, $v$’s neighbors within distance $\tau$ in $S$. For example, in Figure 2(c), the context node of $A_3$ ($\tau = 2$) in the sample walk are $V_1$, $P_2$, $P_4$ and $V_4$. Finally, for each $v$, we sample a negative node $v_m'$ with the same node type of $v_c$ according to $P_{v_m'}(v_c) \propto d_{v_c}^{\gamma}$, where $d_{v_c}$ is the frequency of $v_c$ in $S$. Furthermore, we design a minibatch based Adam Optimizer [8] to train the model. Specifically, at each iteration, we sample a mini-batch of triplets in $T_{walk}$ and accumulate the objective according to equation (8) or (10), then update the parameters via Adam. We repeat the training iterations until the change between two consecutive iterations is sufficiently small. Figure 2(c) shows an illustration of the framework of HSG-SR and SE-HSG on the academic network. The output representations $\Theta$ and $E$ can be utilized in various HetNet mining tasks, as we will show in Section 4.

3.4 Online Update for New Nodes (C3)

The optimization and training strategies in previous section learn representations of all existing nodes but they cannot be employed in an online situation. Considering the growing property of HetNets, we aim to design an online update module to efficiently learn representation of each new node and address challenge C3.

3.4.1 Representation of New Node with Unstructured Semantic Content. As we introduce deep architecture $f$ to encode semantic representations of nodes with unstructured content, the optimized parameters $\Phi^*$ can be directly applied to infer representation of such new node without an extra training step. That is, $E_{v'} = f^*(Y_{v'})$ for new node $v'$, where $Y_{v'}$ is the text content of $v'$ and $f^*$ is the learned semantic encoder with $\Phi^*$. For example, in the academic network, we can use the learned paper semantic encoder to infer the representation of each new paper.

3.4.2 Representation of New Node without Unstructured Semantic Content. Inspired by eALS [6] for online recommendation, we make a reasonable assumption that each new node should not change the learned node representations too much from a global perspective. Accordingly, for each incoming node $v'$ without unstructured content, we leverage objective $o_3$ to formulate the following objective:

$$
o_4 = \sum_{(v', v_c, v_m') \in T_{walk}^{v'}} \log \sigma(\Theta_{v_c}^* \cdot \Theta_{v'}) + \log \sigma(-\Theta_{v_m'} \cdot \Theta_{v'})
$$

(11)

where $\Theta^*$ denotes representation of $v'$, $\Theta^*$ is the learned node representations via SE-HSG, i.e., $\Theta_{v_c}^* = \Theta_{v_c} = f^*(Y_v)$ for $v \in V_5$ otherwise $\Theta_{v_c}^* = \Theta_{v_c}$. That is, we take $\Phi^*$ as constant values in $o_4$. In addition, we leverage meta-path walks rooted at $v'$ to collect triplet set $T_{walk}^{v'}$ for the training process of update module. Specifically, Figure 3 gives an illustrative example of online update for new author in academic network.

![Figure 3: An Illustrative example of online update for new author in academic network.](image-url)
consecutive iterations is sufficiently small. The proposed update module is efficient since only simple meta-path walk sampling and limited update steps on new node representation are performed. In the experimental evaluation, we only consider new author who writes new paper at a venue that exists in network, making the “APVPA” meta-path walk always feasible.

4 EXPERIMENTS

In this section, we conduct extensive experiments with the aim of answering the following research questions:

- (RQ1) How does CARL perform vs. state-of-the-art network representation learning models for different HetNet mining tasks, such as link prediction (RQ1-1), document retrieval (RQ1-2) and node recommendation (RQ1-3)? In addition, how do hyper-parameters impact CARL’s performance in each task?
- (RQ2) What is the performance difference between CARL and baselines in relevance search w.r.t. each task in RQ1?
- (RQ3) What is the performance of CARL’s online update module on the task for new nodes such as category assignment?

Notice that, although our model can be applied to or modified for different HetNets, we focus on experiments on the academic HetNet due to data availability.

4.1 Experimental Setup

4.1.1 Data. We use the AMiner computer science dataset [21], which is publicly available. To avoid noise, we remove the papers published in venues (e.g., workshop) with limited publications and the instances without abstract text. In addition, topic of each area changes over time. For example, according to our analysis of the data, the most popular topics in data mining change from web mining and clustering (1996~2005) to network mining and learning (2006~2015). To make a thorough evaluation of CARL and verify its effectiveness for networks in different decades, we independently conduct experiments on two datasets, i.e., AMiner-I (1996~2005) and AMiner-II (2006~2015). As a result, AMiner-I contains 160,713 authors, 111,409 papers and 150 venues, AMiner-II contains 571,693 authors, 483,449 papers and 492 venues. The structure of the academic network used in this work is shown in Figure 1.

4.1.2 Comparison Baselines. We compare CARL with four state-of-the-art models, i.e., Deepwalk [12], LINE [20], node2vec [5] and metapath2vec [4]. Notice that, we use either random walk (rw) or meta-path walk (mw) to collect context node in HSG-SR and SE-HSG, resulting in four variants of CARL: CARL\textsubscript{rw}, CARL\textsubscript{mw}, CARL\textsubscript{SE-HSG-SR} and CARL\textsubscript{SE-HSG}.

4.1.3 Reproducibility. For fairness comparison, we use the same representation dimension $d = 128$ for all models. The window size $r = 7$, the number of walks per node $N = 10$ and the walk length $L = 30$ are used for Deepwalk, node2vec, metapath2vec and CARL. The same set of parameters is used for CARL’s online update module. The size of negative samples $M$ is set to 5 for node2vec, LINE and metapath2vec. In addition, $y = 1.0$ for CARL\textsubscript{HSG-SR} and three meta-path schemes “APA”, “APPA” and “APVPA” are jointly used to generate meta-path walks for CARL\textsubscript{mw}. We employ TensorFlow to implement all variants of CARL and further conduct them on NVIDIA TITAN X GPU. Code will be available upon publication.

4.2 Link Prediction (RQ1-1)

Who will be your academic collaborators? As a response to RQ1-1, we design an experiment to evaluate CARL’s performance on the author collaboration link prediction task.

4.2.1 Experimental Setting. Unlike past work [5] that randomly samples a portion of links for training and uses the remaining for evaluation, we consider a more realistic setting that splits training/test data via a given time stamp $T$. Specifically, first, the network before $T$ is utilized to learn node representations. Then, the collaboration links before $T$ are used to train a binary logistic classifier. Finally, the collaboration relations after $T$ with equal number of random non-collaboration links are used to evaluate the trained classifier. In addition, only new collaborations among current authors (who appear before $T$) are considered and duplicated collaborations are removed from evaluation. For example, in Figure 1(b), A1 and A4 co-author a new paper $P_4$ after $T$. The classifier tends to predict new collaboration between $A_1$ and $A_4$ using previous links, e.g., collaboration between $A_1$ and $A_3$. The representation of link is formed by element-wise multiplication between representations of two end nodes. We use Accuracy and F1 score of binary classification as the evaluation metrics. Besides, $T$ is set as 2003/2004 and 2013/2014 for AMiner-I and AMiner-II, respectively.

4.2.2 Results. The performances of different models are reported in Table 1. According to the table: (a) All variants of CARL perform better than baselines, demonstrating the effectiveness of incorporating unstructured semantic content to learn author representations; (b) CARL\textsubscript{SE-HSG} outperforms the best performances in all cases. The average improvements of CARL\textsubscript{mw} over different baselines range from 10.9% to 41.0% and 6.7% to 30.9% on AMiner-I and AMiner-II, respectively. (c) CARL\textsubscript{mw} outperforms CARL\textsubscript{rw}, showing that meta-path walk is better than random walk for collecting context node in CARL. In addition, CARL\textsubscript{SE-HSG} has better performance than CARL\textsubscript{HSG-SR}, indicating that concatenating text encoder with heterogeneous SkipGram is more significant than taking text encoding as semantic regularization.

4.2.3 Parameter Sensitivity. We conduct experiment to analyze the impact of two key parameters, i.e., the window size $r$ of walk sampling and the representation dimension $d$. We investigate a specific parameter by changing its value and fixing the others. The prediction results of CARL\textsubscript{mw} as a function of $r$ and $d$ on AMiner-II ($T = 2013$) are shown in Figure 4. We see that: (a) With increasing of $r$, accuracy and F1 score increase at first since a larger window means more useful context information. But when $r$ goes beyond a certain value, performances decrease slowly with $r$ possibly due to uncorrelated noise. The best $r$ is around 7; (b) Similar to $r$, an appropriate value should be set for $d$ such that the best node representations are learned. The optimal $d$ is around 128.

4.3 Document Retrieval (RQ1-2)

Which relevant papers should be retrieved for your query? As a response to RQ1-2, we design an experiment to evaluate CARL’s performance on the paper retrieval task.
that semantic enhanced SkipGram is better than SkipGram with semantic regularization. However, CARL\textsuperscript{SE} has performance close to that of CARL\textsuperscript{RW}, indicating that meta-path walk has little impact for this task. It is reasonable since the paper representations depend on the deep semantic encoder in our model.

4.3.3 Parameter Sensitivity. Following the same setup in link prediction, we investigate the impact of window size $\tau$ and representation dimension $d$ on CARL\textsuperscript{RW}'s performance on AMiner-II (T = 2013), as shown by Figure 5. It can be seen that: (a) The results are little sensitive to $\tau$ when $\tau \geq 7$. As we noted above, paper representations depend on the deep semantic encoder; (b) The dimension $d$ plays significant role on generating paper representations. The best representations are learned when $d$ is around 128 for the paper retrieval task.

4.4 Node Recommendation (RQ1-3)
Which venues should be recommended to you? As a response to RQ1-3, we design an experiment to evaluate CARL's performance on the venue recommendation task.

4.4.1 Experimental Setting. As in the previous two tasks, the network before $T$ is utilized to learn node representations. The ground truth of recommendation is based on author's appearance in venue after $T$. For example, in Figure 1(b), author $A_1$ writes:

**Table 1:** Collaboration prediction results comparison.

| AMiner-I | T = 2004 | T = 2003 | Gain |
|----------|----------|----------|------|
| Accuracy | F1       | Accuracy | F1   |
| Deepwalk | 0.6597   | 0.5024   | 0.6487 | 0.4835 | 30.9% |
| LINE     | 0.7034   | 0.6048   | 0.6956 | 0.5898 | 14.3% |
| node2vec | 0.7136   | 0.6222   | 0.7066 | 0.5965 | 12.7% |
| metapath2vec | 0.7299* | 0.6628* | 0.7254* | 0.6512* | 6.7% |
| CARL\textsuperscript{RW}-SR | 0.7498 | 0.7061 | 0.7495 | 0.6946 | – |
| CARL\textsuperscript{RW}-SE-HSG | 0.7562 | 0.7125 | 0.7546 | 0.6978 | – |
| CARL\textsuperscript{RW}-SE-HSG | 0.7627 | 0.7208 | 0.7602 | 0.7097 | – |

**Table 2:** Paper retrieval results comparison.

| AMiner-I | T = 2003 | T = 2004 | Gain |
|----------|----------|----------|------|
| Accuracy | F1       | Accuracy | F1   |
| Deepwalk | 0.8120   | 0.8816   | 0.8217 | 0.8967 | 6.7% |
| LINE     | 0.7320   | 0.8130   | 0.7485 | 0.8380 | 16.4% |
| node2vec | 0.8552*  | 0.9148*  | 0.8653* | 0.9250* | 2.3% |
| metapath2vec | 0.8239 | 0.8910 | 0.8366 | 0.9081 | 5.3% |
| CARL\textsuperscript{RW}-SE-HSG | 0.8685 | 0.9351 | 0.8696 | 0.9375 | – |
| CARL\textsuperscript{RW}-SE-HSG | 0.8673 | 0.9342 | 0.8722 | 0.9389 | – |
| CARL\textsuperscript{RW}-SE-HSG | 0.8741 | 0.9412 | 0.8788 | 0.9455 | – |
| CARL\textsuperscript{RW}-SE-HSG | 0.8751 | 0.9425 | 0.8783 | 0.9470 | – |

**Figure 4:** Parameter sensitivity in collaboration prediction.

**Figure 5:** Parameter sensitivity in paper retrieval.

(a) Window size $\tau$
(b) Representation dimension $d$
Table 3: Venue recommendation results comparison.

| AMiner-I                  | T = 2004 | T = 2003 | Gain     |
|---------------------------|----------|----------|----------|
|                            | Rec@5    | Rec@10   | Rec@5    | Rec@10   |
| Deepwalk                  | 0.1051*  | 0.1628*  | 0.0864*  | 0.1403*  | 18.9%    |
| LINE                      | 0.0376   | 0.0677   | 0.0388   | 0.0717   | 178.7%   |
| node2vec                  | 0.0945   | 0.1570   | 0.0774   | 0.1386   | 27.2%    |
| metapath2vec              | 0.0878   | 0.1527   | 0.0714   | 0.1395   | 33.3%    |
| CARL<sup>sw</sup>−SE−HSG  | 0.1156   | 0.1772   | 0.0988   | 0.1593   | –        |
| CARL<sup>mw</sup>−SE−HSG  | 0.1208   | 0.1813   | 0.1050   | 0.1631   | –        |
| CARL<sup>sw</sup>−HSG−SR  | 0.1225   | 0.1831   | 0.1054   | 0.1648   | –        |
| CARL<sup>sw</sup>−HSG−SE−HSG | 0.1250 | 0.1852   | 0.1073   | 0.1667   | –        |

| AMiner-II                 | T = 2014 | T = 2013 | Gain     |
|---------------------------|----------|----------|----------|
|                            | Rec@5    | Rec@10   | Rec@5    | Rec@10   |
| Deepwalk                  | 0.1039*  | 0.1549*  | 0.0925*  | 0.1384*  | 11.7%    |
| LINE                      | 0.0297   | 0.0470   | 0.0267   | 0.0433   | 275.6%   |
| node2vec                  | 0.0766   | 0.1182   | 0.0691   | 0.1074   | 47.8%    |
| metapath2vec              | 0.0654   | 0.1077   | 0.0608   | 0.1001   | 66.1%    |
| CARL<sup>sw</sup>−SE−HSG  | 0.1092   | 0.1643   | 0.0977   | 0.1466   | –        |
| CARL<sup>sw</sup>−HSG−SR  | 0.1116   | 0.1713   | 0.1032   | 0.1526   | –        |
| CARL<sup>sw</sup>−HSG−SE−HSG | 0.1107 | 0.1725   | 0.1023   | 0.1534   | –        |
| CARL<sup>sw</sup>−SE−HSG  | 0.1136   | 0.1742   | 0.1045   | 0.1551   | –        |

4.5 Relevance Search: Case Study (RQ2)

To answer RQ2, we present three case studies of relevance search on AMiner-II (T = 2013) to show the performance differences between CARL<sup>sw</sup>−SE−HSG and baselines. The ranking of each search result is based on the cosine similarity of representations.

4.5.1 Relevant Author Search. Table 4 lists the top-5 returned authors for query author "Jure Leskovec" of CARL<sup>sw</sup>−SE−HSG and two baselines, i.e., node2vec and metapath2vec, which achieve relatively better performances in the collaboration prediction task. According to this table: (a) most of returned authors of node2vec have collaboration relations with "Jure" before T, indicating that node2vec highly depends on structural closeness and cannot find relevant authors who are far away from "Jure" in the network; (b) metapath2vec returns some authors (e.g., P. Nguyen) who are different from "Jure" in their specific research interests, illustrating that "APVPA" meta-path walks (used by metapath2vec) may collect context node that are different from target node since it is common that authors bridged by the same venue study different research topics; (c) CARL<sup>sw</sup>−SE−HSG not only returns structurally close authors who have collaboration relations with "Jure" before T but also finds farther authors (e.g., D. Romero) who share similar research interest with "Jure", demonstrating CARL<sup>sw</sup>−SE−HSG captures both structural closeness and unstructured semantic relations for learning author representations.

4.5.2 Relevant Paper Search. Table 5 lists the top-5 returned papers for query paper "When will it happen?" of CARL<sup>sw</sup>−SE−HSG and the best baseline node2vec in paper retrieval task. From this table: (a) all returned papers of node2vec are written by at least one author in query paper, showing that node2vec only returns structurally close papers but has difficulty finding farther semantically related papers; (b) CARL<sup>sw</sup>−SE−HSG not only returns structurally close papers which have common authors with query paper but also finds semantically related papers without authorship overlapping, showing that CARL<sup>sw</sup>−SE−HSG utilizes both structural content and unstructured semantic content for learning paper representations.

4.5.3 Relevant Author-Venue Search. Table 6 lists the top-10 returned venues for query author "Chi Wang" of CARL<sup>sw</sup>−SE−HSG and two baselines, i.e., Deepwalk and node2vec, which have relatively better performances on the venue recommendation task. We find that: (a) Deepwalk and node2vec recommend both data mining (e.g., KDD & ICDM) and database (e.g., SIGMOD & PVLDB) venues to "Chi" since some of his works cite database papers and some of his co-authors focus on database research, which illustrates that both Deepwalk and node2vec return structurally close venues, some of which are not the most suitable ones; (b) most of the venues in CARL<sup>sw</sup>−SE−HSG's recommendation list belong to data mining related areas, demonstrating that incorporating semantic content helps learn better representations of author and venue.

4.6 Category Visualization of New Nodes (RQ3)

The previous experiments and case studies demonstrate the effectiveness of CARL in learning representations of current nodes, i.e., nodes that exist in HetNet before T. As described in Section 3.4, the learned semantic encoder of CARL can be directly applied to infer representation of each incoming node with semantic content.
Table 4: Case study of relevant author search. "Coauthor-b" denotes whether two authors have a collaboration relation before T and "Similar-I" represents whether two authors have similar research interests.

| Rank | node2vec | metaphor2vec | CARL<sub>SE-HSG</sub> |
|------|----------|--------------|------------------------|
|      | Author   | Coauthor-b? | Similar-I? | Author | Coauthor-b? | Similar-I? | Author | Coauthor-b? | Similar-I? |
| 1    | S. Kairam | ✓           | ✓          | L. Backstrom | ✓           | ✓          | J. Kleinberg | ✓           | ✓          |
| 2    | M. Rodriguez | ✓         | ✓          | P. Nguyen | ✓           | ✓          | D. Romero | ✓           | ✓          |
| 3    | D. Wang | ✓           | ✓          | S. HanhijÃďrvi | ✓ | ✓ | A. Dasgupta | ✓ | ✓ |
| 4    | J. Yang | ✓           | ✓          | S. Myers | ✓           | ✓          | L. Backstrom | ✓ | ✓ |
| 5    | A. Jaimes | ✓           | ✓          | V. Lee | ✓           | ✓ | G. Kossinets | ✓ | ✓ |

Table 5: Case study of relevant paper search.

| Model       | Rank | Returned Paper                                                                 |
|-------------|------|-------------------------------------------------------------------------------|
| node2vec    | 1    | Co-author relationship prediction in heterogeneous bibliographic networks (ASONAM2011), A: Y. Sun, R. Barber, M. Gupta, C. Aggarwal, J. Han |
|             | 2    | A framework for classification and segmentation of massive audio data streams (KDD2007), A: C. Aggarwal |
|             | 3    | Mining heterogeneous information networks: the next frontier (KDD2012), A: J. Han |
|             | 4    | Ranking-based classification of heterogeneous information networks (KDD2011), A: M. Ji, J. Han, M. Danilevsky |
|             | 5    | Evolutionary clustering and analysis of bibliographic networks (ASONAM2011), A: M. Gupta, C. Aggarwal, J. Han, Y. Sun |
| CARL<sub>SE-HSG</sub> | 1    | Collective prediction of multiple types of links in heterogeneous information networks (ICDM2014), A: B. Cao, X. Kong, F. Yu |
|             | 2    | Community detection in incomplete information networks (WWW2012), A: W. Lin, X. Kong, P. Yu, Q. Wu, Y. Jia, C. Li |
|             | 3    | Meta path-based collective classification in heterogeneous information networks (CIKM2012), A: C. Kong, P. Yu, Y. Ding, D. Wild |
|             | 4    | Ranking-based classification of heterogeneous information networks (KDD2011), A: M. Ji, J. Han, M. Danilevsky |
|             | 5    | Fast computation of SimRank for static and dynamic information networks (EDBT2010), A: C. Li, J. Han, G. He, X. Jin, Y. Sun, Y. Yu, T. Wu |

Table 6: Case study of relevant author-venue search.

| Model       | Rank | Returned Paper                                                                 |
|-------------|------|-------------------------------------------------------------------------------|
| Deepwalk    | 1    | KDD                                                                            |
|             | 2    | CIKM                                                                           |
|             | 3    | SIGMOD                                                                         |
|             | 4    | SIGMOD                                                                         |
|             | 5    | ICDM                                                                           |
|             | 6    | WWW                                                                            |
|             | 7    | ICDE                                                                           |
|             | 8    | PVLDB                                                                          |
|             | 9    | WWW                                                                            |
|             | 10   | EDBT                                                                           |

Besides, CARL's online update module can efficiently learn representation of each new node without semantic content. To answer RQ3 and show the effectiveness of learned new node representations, we employ the Tensorflow embedding projector to visualize the new paper representations inferred by paper semantic encoder and new author representations learned by online update module in sequence. Figure 7 shows the results of new paper and author nodes (appearing after T) of four selected research categories, i.e., Data Mining (DM), Computer Vision (CV), Human Computer Interaction (HCI) and Theory, on AMiner-II (T = 2013). Specifically, we choose three top venues for each area. Each new paper is assigned according to venue's area and the category of each author.

is assigned to the area with the majority of his/her publications. We randomly sample 100 new papers/authors of each area. According to Figure 7: (a) The representations of new papers in the same category cluster closely and can be well discriminated from others for both 2D and 3D visualizations, indicating that the semantic
CARL: Content-Aware Representation Learning for Heterogeneous Networks

5 RELATED WORK

In the past decade, many works have been devoted to mining HetNets for different applications, such as relevance search [2, 7, 16, 28], node clustering [17, 18], personalized recommendation [14, 26, 27].

The network representation learning has gained a lot of attention in the last few years. Some walk sampling based models [4, 5, 12] have been proposed to learn vectorized node representations that can be further utilized in various tasks in network. Specifically, inspired by word2vec [10] for learning distributed representations of words in text corpus, Perozzi et al. developed the innovative Deepwalk [12] which introduces node-context concept in network (analogous to context-word) and feeds a set of random walks over models by using time series information of nodes and links. Sign the dynamic heterogeneous network representation learning category visualization of new nodes. In the future, we plan to design the online update module in learning new node author representations.

6 CONCLUSION

In this paper, we formalize the problem of content-aware representation learning in HetNets and propose a novel model CARL to solve the problem. CARL performs joint optimization of heterogeneous SkipGram and deep semantic encoding for capturing both structural closeness and unstructured semantic relations in a HetNet. Furthermore, an online update module is designed to efficiently learn representations of new nodes. Extensive experiments demonstrate that CARL outperforms state-of-the-art baselines in various HetNet mining tasks, such as link prediction, document retrieval, node recommendation and relevance search. Besides, the online update module achieves satisfactory performance, as reflected by category visualization of new nodes. In the future, we plan to design the dynamic heterogeneous network representation learning models by using time series information of nodes and links.

REFERENCES

[1] Shiyu Chang, Wei Han, Jiiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. 2015. Heterogeneous network embedding via deep architecture. In KDD. 119–128.
[2] Ting Chen and Ziyun Sun. 2017. Task-Guided and Path-Augmented Heterogeneous Network Embedding for Author Identification. In WSDM. 295–304.
[3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:1406.1078 (2014).
[4] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In KDD. 135–144.
[5] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In KDD. 855–864.
[6] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In SIGIR. 549–558.
[7] Zhipeng Huang, Yudian Zheng, Reynold Cheng, Yizhou Sun, Nikos Mamoulis, and Xiang Li. 2016. Meta structure: Computing relevance in large heterogeneous information networks. In KDD. 1595–1604.
[8] Diederik F Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
[9] Xiaozhong Liu, Yingying Yu, Chun Guo, and Yizhou Sun. 2014. Meta-path-based ranking with pseudo relevance feedback on heterogeneous graph for citation recommendation. In CIKM. 121–130.
[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In NIPS. 3115–3119.
[11] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric metric transitivity preserving graph embedding. In KDD. 1105–1114.
[12] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In KDD. 701–710.
[13] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec. In WSDM.
[14] Xiang Ren, Jialiu Liu, Xiao Yu, Urvashi Khandelwal, Quanaquan Gu, Lidan Wang, and Jiawei Han. 2014. Clucite: Effective citation recommendation by information network-based clustering. In KDD. 821–830.
[15] Yizhou Sun, Jiawei Han, Charu C Aggarwal, and Nitesh V Chawla. 2012. When will it happen?: relationship prediction in heterogeneous information networks. In WSDM. 663–672.
[16] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In VLDB. 4, 11 (2011), 992–1003.
[17] Yizhou Sun, Brandon Norick, Jaewei Han, Xifeng Yan, Philip Yu, and Xiao Yu. 2012. PathSelClus: Integrating Meta-Path Selection with User-Guided Object Clustering in Heterogeneous Information Networks. In WSDM. 1348–1356.
[18] Yizhou Sun, Yintao Yu, and Jiawei Han. 2009. Ranking-based clustering of social representations. In KDD. 701–710.
[19] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric metric transitivity preserving graph embedding. In KDD. 1105–1114.
[20] Tianyi Wu, Yintao Yu, and Jiawei Han. 2009. Ranking-based clustering of social representations. In KDD. 701–710.
[21] Tye Wang, Jiawei Han, Yantao Jia, Jie Tang, Duo Zhang, Yantao Yu, and Jingyi Guo. 2010. Mining advisor-advisee relationships from research publication networks. In KDD. 203–212.
[22] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In KDD. 1225–1234.
[23] Hongwei Wang, Jia Wang, Jialing Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. GraphGAN: Graph Representation Learning with Generative Adversarial Nets. In AAAI.
[24] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. 2017. Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation. In KDD. 1245–1254.
[25] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In WSDM. 283–292.
[26] Chuxu Zhang, Chao Huang, Lu Yu, Xiangliang Zhang, and Nitesh Chawla. 2018. Camel: Content-Aware and Meta-path Augmented Metric Learning for Author Identification. In WWW.