Fine-grained Event Categorization with Heterogeneous Graph Convolutional Networks

Hao Peng[1,2], **Jianxin Li**[1,2], **Qiran Gong**[1,2], **Yangqiu Song**[3], **Yuanxing Ning**[1,2],
Kunfeng Lai[4] and **Philip S. Yu**[5,6]

[1] Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University
[2] State Key Laboratory of Software Development Environment, Beihang University
[3] Department of Computer Science and Engineering, Hong Kong University of Science and Technology
[4] Platform and Content Group, Tencent
[5] Institute for Data Science, Tsinghua University
[6] Department of Computer Science, University of Illinois at Chicago
{penghao, lijx, ningyx}@act.buaa.edu.cn, allen_gong@buaa.edu.cn, yqsong@cse.ust.hk,
calvinlai@tencent.com, psyu@uic.edu

Abstract

Events are happening in real-world and real-time, which can be planned and organized occasions involving multiple people and objects. Social media platforms publish a lot of text messages containing public events with comprehensive topics. However, mining social events is challenging due to the heterogeneous event elements in texts and explicit and implicit social network structures. In this paper, we design an event meta-schema to characterize the semantic relatedness of social events and build an event-based heterogeneous information network (HIN) integrating information from external knowledge base, and propose a novel Pairwise Popularity Graph Convolutional Network (PP-GCN) based fine-grained social event categorization model. We propose a Knowledgeable meta-paths Instances based social Event Similarity (KIES) between events and build a weighted adjacent matrix as input to the PP-GCN model. Comprehensive experiments on real data collections are conducted to compare various social event detection and clustering tasks. Experimental results demonstrate that our proposed framework outperforms other alternative social event categorization techniques.

1 Introduction

Events are happening in real-world and real-time, which can be planned and organized occasions involving multiple people and objects, such as a social gathering, celebrity activities or a sports competition in some specific location at a particular time. Nowadays, social media platforms have become major sources for publicizing events. Events announced on social media usually attract comments and reposts with opinions and emotions, and such content can reflect public opinion about many social, political, economic issues, etc. Mining of social media posts, such as fine-grained social event categorization, will benefit a lot of real applications, such as information organization, predictive analysis, disaster risk analysis, and others [Atefeh and Khreich, 2015; Aggarwal and Subbian, 2012; Allan, 2012]. In general, fine-grained social event categorization focus on event detection and event clustering.

The tasks of fine-grained social event categorization are more challenging than traditional text mining or social network mining, since social event is a combination of social network and the information flows (in terms of short messages) over it. On the one hand, modeling social events is very complicated and ambiguous. Social events are described in short texts and usually contain different types of entities, such as person, location, organization, number, time, etc [Allan, 2012; Ji and Grishman, 2008; Yu *et al.*, 2017]. Moreover, events are commented or retweeted by social network users. Thus, modeling social event needs to consider heterogeneous elements as well as explicit and implicit social network structures within social posts. On the other hand, models of fine-grained event categorization often have bottlenecks in which the number of the categories is large and the number of samples per class is small. Thus, fine-grained event categorization needs to address the accuracy of the developed algorithms. Currently, fine-grained text classification is more difficult and lacks related research work than the fine-grained object recognition in other fields such as computer vision [Zhang *et al.*, 2014].

A handful of studies [Ritter *et al.*, 2012; Chandola *et al.*, 2009; Becker and Gravano, 2011; Shao *et al.*, 2017] have investigated leveraging homogeneous graphs or manually defined frames for social event modeling and extracting. The first line of thought is to treat social event as homogeneous words/elements co-occurrence graph [Chandola *et al.*, 2009; Aggarwal and Subbian, 2012; Angel *et al.*, 2012; Liu *et al.*, 2019]. Typically, they construct a homogeneous words/elements co-occurrence graph, and then consider different scales of abnormally connected subgraph structures

(under different names such as k-clique, motifs or graphlets) as the social events. Despite the compelling results achieved by these studies, their categorization accuracies remain unsatisfactory for building reliable and open domain event detection and clustering systems in practice. The second line of thought is to use manually defined frames-based event definitions applying the well-defined techniques for extracting social event frames from news [Kim *et al.*, 2009; Ji and Grishman, 2008]. The frame-based event extraction can extract entities and their relationships, but uses only a limited number of event types, such as earthquake disaster, stock market, venues, politics, etc. Moreover, it uses complicated machine learning models, usually a pipeline of them, to incorporate different levels of annotation and features.

Social media events can be regarded as a co-occurrence of event elements including themes, dates, locations, people, organizations, keywords and social behavior participants. The simplest way to monitor social media events is to represent events as bags-of-words, but it will be more semantically meaningful if we can annotate words and multi-word-expressions as entities with types. For example, in the tweet "*China Seismological Network: The earthquake struck at 21:19:46 China Standard Time on 8 August 2017 in Zhangzha Town in Jiuzhaigou County with magnitude 7.0*", there are multiple event elements: **Time**: *21:19:46*; **Date**: *August 8, 2017*; **Timezone**: *China Standard Time*; **Town**: *Zhangzha*; **County**: *Jiuzhaigou*; **Nation**: *China*; **Magnitude**: *7.0*; **Poster**: *China Seismological Network*. Obviously, the above event's elements are of different types. Moreover, in addition to intuitive co-occurrence relationship, after extracting entities, we can make use of external knowledge base [Auer *et al.*, 2007; Xu *et al.*, 2017] to complement more relationships between entities, such as "**located-in**" relationships with other locations, "**attribute-of**" relationships with magnitude and earthquake, etc. Thus, a message mentioning an event can be related to its keywords, entities (and their relations), topics, etc. Furthermore, the social network users posting messages are also connected with different relationships, such as following/followed and retweeting. Thus we can model social media events as HIN [Shi *et al.*, 2017].

In this paper, we first present event instance (shown in short text message) as hyper-edge in an HIN, where all the keywords, entities, topics and social users can be connected by this hyper-edge, and define an event meta-schema to characterize the semantic relatedness of social event instances and build event-based HIN. In order to enrich the HIN, we extract some information as a complement of the relationships based on the external knowledge base and algorithms. Based on the event HIN, we define a weighted <u>K</u>nowledgeable meta-paths <u>I</u>nstances based <u>E</u>vent <u>S</u>imilarity measure, namely KIES, from semantically meaningful meta-paths. In order to accurately measure the weights between meta-paths and perform fine-grained event detection, we then design a novel <u>P</u>airwise <u>P</u>opularity <u>G</u>raph <u>C</u>onvolutional <u>N</u>etwork model, namely PP-GCN, to learn the representation of each event instance. Finally, under the HINs-based event modeling, we present a KIES-measure based fine-grained event clustering.

Compared to traditional methods, the proposed models have several advantages: (1) By modeling social events based on a HIN, the proposed framework can integrate event elements, such as keywords, topic, entities, social users and their relations, in a semantically meaningful way, and can also calculate the similarity between any two event instances. (2) By modeling pairwise popularity graph convolutional network, the model achieves state-of-the-art results and avoids overfitting in fine-grained event detection tasks. (3) The proposed KIES with learned weights between meta-paths by the PP-GCN can boost the performance of fine-grained social events clustering compared to existing state-of-the-art baselines methods. The code of this work is publicly available at *https://github.com/RingBDStack/PPGCN*.

2 Heterogeneous Event Modeling

In this section, we define the problem of modeling social events in heterogeneous information network (HIN) and introduce several related concepts and necessary notations.

2.1 Event Modeling in HIN

The definition and characterization of "social event" have received substantial attention across academic fields, from language [Miller, 1998] to cognitive psychology [Zacks and Tversky, 2001]. A social event generally refers to influential facts that appear on social networks and occur in the real world, including creators (posters), named entities such as participants, organizations, festival, specific times, places, currency, address, etc., and other elements such as keywords and topics. We name the above elements as *event-oriented elements*. However, extracting the event-oriented elements from the original social text message with NLP tools[1],[2] is still a prior processing work. Even within most of the events, there are some relationships between event-oriented elements, such as relationships between entities, relationships between keywords, relationships between topics, explicit and implicit relationships between social users, and so on. We name the above relationships as *event-elements relationships*.
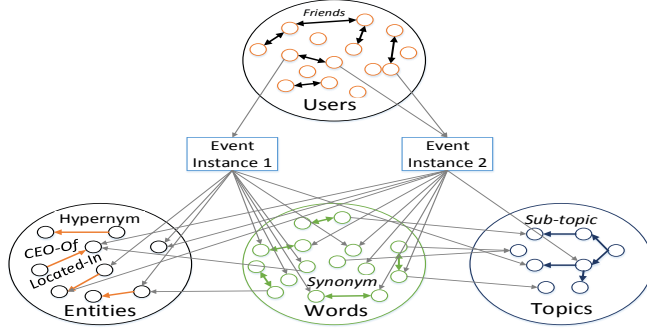
We use the manually organized synonyms[3] to add synonym relationship among keywords in the event-based HIN. For hierarchical topic structures and the affiliation relationship between keywords and topics in the event-based HIN, we employ the hierarchical latent Dirichlet allocation technologies [Griffiths *et al.*, 2004; Blei *et al.*, 2010] based on the existing toolbox[4] (with about 30 most probable words for each topic). In order to build the relationship between entities in the event-based HIN, we consider both accuracy and efficiency, and tackle the problem by following three-steps. First, we retrieve the same entity candidate from knowledge base, such as the Chinses CN-DBpedia [Xu *et al.*, 2017]. Second, we use word embeddings [Mikolov *et al.*, 2013] based Word Mover's Distance technology [Kusner *et al.*, 2015] to measure the similarity between context of entity in the social text and description of entity in the candidate, and choose the entity from the candidate with highest similarity. Third, we
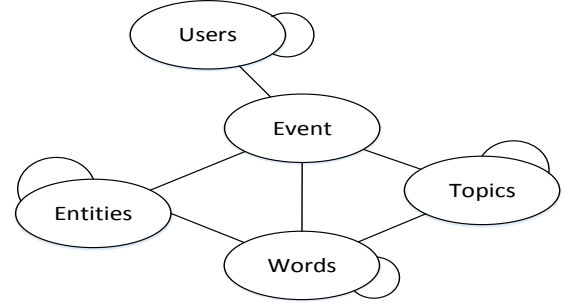
---

[1]https://github.com/stanfordnlp/CoreNLP

[2]https://github.com/NLPIR-team/NLPIR

[3]https://github.com/huyingxi/Synonyms

[4]https://github.com/joewandy/hlda

(a) Example of two event instances connected by different types of nodes and edges

(b) Meta-schema of event-based HIN

Figure 1: Illustration of events as a HIN, where event instances are represented as hyper-edges.

query the relationship between aligned entities in the knowledge base as the final relationship of entities in *event-elements relationships*. In order to establish the relationship between entities and keywords in the event-based HIN, we extract the keywords in the relevant description of each entity in the knowledge base and use this affiliation as the relationship between the entity and the keywords. For the relationship between social users, we consider users with a large number of friends, and store the relationship between users in advance.

After extracting the above event-oriented elements and event-elements relationships from event instances, we build an event-based HIN, as shown in Figure 1(a). The social event can be regarded as a co-occurrence of event-oriented elements, and the event-elements relationships are conducive to explaining the relationship between various elements. Thus, an event instance can be treated as a subgraph of the whole HIN. One particular advantage of the HIN is that meta-paths defined over types (e.g., a typical meta-path "event-entity-event" represents the event similarity based on overlapped entities between two event instances) can reflect semantically meaningful information about similarities, and thus can naturally provide explainable results for event modeling.

2.2 Preliminaries

We introduce some basic definitions from previous works [Sun and Han, 2013; Shi *et al.*, 2017], and give some event-HIN examples.

Definition 2.1 *A **heterogeneous information network (HIN)** is a graph $G = (V, E)$ with an entity type mapping $\phi : V \to A$ and a relation type mapping $\psi : E \to R$, where V denotes the entity set, E denotes the link set, R denotes the relation type set and A denotes the entity type set. The number of entity types $|A| > 1$ or the number of relation types $|R| > 1$.*

For example, Figure 1(a) shows an example of two event instances connected with different types of entities, keywords, topics, social users and relationships. After giving a complex HIN for event modeling, it is necessary to provide its meta level (i.e., schema-level) description for better understanding.

Definition 2.2 *Given an HIN $G = (V, E)$ with the entity mapping $\phi : V \to A$ and the relation type mapping $\psi :$*

$E \to R$, *the **meta-schema** (or network schema) for network G, denoted as $T_G = (A, R)$, is a graph with nodes as entity types from A and edges as relation types from R.*

For example, Figure 1(b) shows an example of the HIN meta-schema characterizing events on social messages. Another important concept is the meta-path which systematically defines relationships between entities at the schema level.

Definition 2.3 *A **meta-path** P is a path defined on the graph of network schema $T_G = (A, R)$ of the form $A_I \xrightarrow{R_1} A_2 \xrightarrow{R_2} A_3 \cdots A_L \xrightarrow{R_L} A_{L+1}$ which defines a composite relation $R = R_1 \cdot R_2 \cdots \cdots R_L$ between objects $A_1, A_2, A_3 \cdots A_{L+1}$, where $\cdot$ denotes relation composition operator, and $L+1$ is the length of P.*

For simplicity, we use object types connected by $\to$ to denote the meta-path when there are no multiple relations between a pair of types: $P = (A_1 - A_2 - \cdots - A_{L+1})$. We say that a meta-path instance $p = (v_1 - v_2 - \cdots - v_{L+1})$ between v_1 and v_{L+1} in network G follows the meta-path P, if $\forall l$, $\phi(v_l) = A_l$ and each edge $e_l = < v_l, v_{l+1} >$ belongs to each relation type $R_l \in P$. We call these paths as path instances of P, denoted as $p \in P$. R_l^{-1} represents the reverse order of relation R_l. We will introduce more semantically meaningful meta-paths that describe event relations in next section.

3 The Proposed Model

In this section, we introduce definitions about knowledgeable meta-paths instances based event similarity measure, and present the technical details about Pairwise Popularity GCN.

3.1 Event Similarity Measure

Before definite the social event similarity, we first present the definition of *CouP* as following,

Definition 3.1 *CouP: Given a meta-path $P = (A_1 - A_2 \cdots A_{L+1})$, CouP is a function of the count of meta-path instances such that $CouP_P(v_i, v_j) = M_P(v_i, v_j)$ where $M_P = W_{A_1 A_2} \cdot W_{A_2 A_3} \cdots W_{A_L A_{L+1}}$ and $W_{A_k A_{k+1}}$ is the adjacency matrix between types A_k and A_{k+1} in the meta-path P.*

For example, for event instance similarity based on event-oriented elements and event-element relationships, the composite relation of two event instances containing the same event element and co-occurrence relationship can be described as "event Instance - Element - event Instance (IEI)" for simplicity. This meta-path simply gives us $M_{IEI} = W_{IE}W_{EI}^T$, which is the dot product between event instances, where W_{EI} is the event Instance-Element co-occurrence matrix. The similarity based on this meta-path instances is accurate because different elements and lengths are considered. The more meta-paths enumerated by the meta-schema, the higher accuracy of the similarity metric is. We can give more event related meta-paths over different lengths, e.g., P_1: *Event-(posted by)-Social user-(post)-Event*, P_2: *Event-(having)-Washington DC-(capital of)-United States-(contained by)-Event*, P_3: *Event-(belong to)-Politician-(relevant)-president-(member of)-Ruling Party-(contained by)-Event*, etc. P_1 means two event instances are similar if they are posted by the same social user. P_2 means two event instances are similar if they mention Washington DC and the United States, respectively, where Washington DC is the capital of the United States. P_3 means two event instances are similar if they can be associated by a chain of three event elements with meaningful relationships. Note that the meta-path does not need to satisfy symmetry. Here, we enumerate 22 symmetric meta-paths in the meta-schema of event-based HIN.

However, if the counts are not normalized for different meta-paths, it is difficult to compare over different meta-path-based similarities. Then, similar to the HIN-based document similarity [Wang *et al.*, 2018], we also define our knowledgeable meta-paths instances based social event similarity measure, namely KIES. Intuitively, if two event instances are more strongly connected by the important (i.e., highly weighted) meta-paths, they tend to be more similar.

Definition 3.2 *KIES: a knowledgeable meta-paths instances based social event similarity. Given a collection of meaningful meta-paths, denoted as $\boldsymbol{P} = \{P_m\}_{m=1}^{M'}$, the KIES between two event instances e_i and e_j is defined as:*

$$KIES(e_i, e_j) = \sum_{m=1}^{M'} \omega_m \frac{2 \times CouP_{P_m}(e_i, e_j)}{CouP_{P_m}(e_i, e_i) + CouP_{P_m}(e_j, e_j)}, \tag{1}$$

where $CouP_{P_m}(e_i, e_j)$ is a count of meta-path P_m between event instances e_i and e_j, $CouP_{P_m}(e_i, e_i)$ is that between event instances e_i and e_i, and $CouP_{P_m}(e_j, e_j)$ is that between event instances e_j and e_j. We use a parameter vector $\vec{\omega} = [\omega_1, \omega_2, \ldots, \omega_{M'}]$ to denote the meta-path weights, where ω_m is the weight of meta-path P_m. $KIES(e_i, e_j)$ is defined in two parts: (1) the semantic overlap in the numerator, which is defined by the number of meta-paths between event instances e_i and e_j; and (2) the semantic broadness in the denominator, which is defined by the number of total meta-paths between themselves. Therefore, we can give a KIES distance with weights for any two event instances.

3.2 Pairwise Popularity GCN Model
Next, we show how to implement fine-grained event detection on social message texts through the pairwise popularity GCN
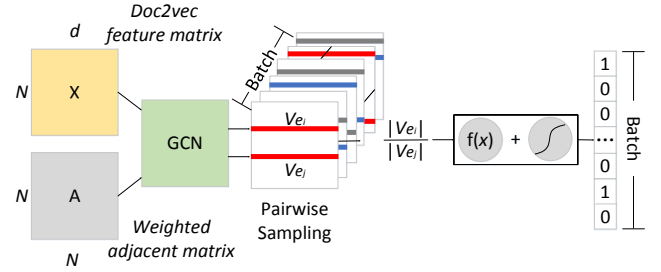

Figure 2: An overview of the proposed Pairwise Popularity Graph Convolutional Network (PP-GCN).

model (PP-GCN), and learn the weights $\vec{\omega}$ for meta-paths, to overcome the problems of a large number of categories and a small number of samples per class.

After computing the distance of any two event instances by the KIES, we can construct a $N \times N$ weighed adjacent matrix A for manually annotated social event instances, where N is the number of event instances and $A_{ij} = A_{ji} = KIES(e_i, e_j)$. Then, we train the Doc2vec [Le and Mikolov, 2014] representation as generalized event instance feature. So, we can construct a $N \times d$ feature matrix X, where d is the dimension of event instance feature. Obviously, so far, we can use the popular GCN [Kipf and Welling, 2017] architecture to learn discriminating event representation based on the interactions among event instances and generalized event instance features in node classification task. The input to the GCN model includes the A and X matrices. Here, one class represents one social event class. In order to construct preliminary GCN model, we utilize the popular multi-layer GCN with the following layer-wise propagation rule [Kipf and Welling, 2017]:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}), \tag{2}$$

where $\widetilde{A} = A + I_N$, $\widetilde{D}$ is diagonal matrix such that $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$ are the adjacency matrix, I_N is the identity matrix, W is the parameter matrix, and l is the number of layers. Let Z be an output $N \times F$ feature matrix, where F is the dimension of output representation per event instance. The input layer to the GCN is $H^{(0)} = X, X \in R^{N \times d}$, which contains original event instance feature, $H^{(l)} = Z$, and Z is graph-level output. And σ denotes an activation function such as Sigmoid or ReLU.

However, the real-world social events naturally have two problems of sparsity: the small number of event instances for each classification and a large number of categories. So, we sample event instances pair and judge whether the pair belongs to one event to train a pairwise GCN model. As shown in Figure 2, we present the proposed PP-GCN model. Before explaining the PP-GCN model, we show how to implement a pairwise sampling to generate training samples. We assume that if a pair of event instances e_i and e_j belongs to the same event classification, we name the pair e_i and e_j as a *positive-pair* sample. If a pair of event instances e_i and e_j belongs to two different events classification, we name the pair e_i and e_j as a *negative-pair* sample. As shown in Figure 2, if the pair is

a positive-pair sample, we represent its by two red lines; if the pair is negative-pair sample, we use both gray line and blue line to represent it. After explaining the training samples, we first randomly select R (i.e., 1000) event instances as a preliminary set, then randomly select two event instances for each event instance in the set to form one positive-pair sample and one negative-pair sample, and finally we can construct a $2R$ event instance pairs set from training samples. Here, both the positive-pair and negative-pair samples are equal to R. Second, we randomly sample the B (i.e., 64) samples from the $2R$ (i.e., 2000) event instance pairs set to form a batch to forward propagation of our proposed model. Third, the second step is cycled E (i.e., 32) times to form an epoch. For next epoch, we loop through the above three steps.

However, the above pairwise sampling based GCN model can not guarantee that the model avoids over-fitting during training. Suppose that any event classification has an average of r event instances, the probability that any event instance selected into a positive-pair sample is $\frac{1}{r}$, and the probability of being selected into a negative-pair sample is about $\frac{1}{N-r}$. We note that $\frac{1}{r} \gg \frac{1}{N-r}$ in general. Obviously, the negative-pair samples have more diversity than the positive-pair samples. [Papadopoulos *et al.*, 2012] has observed the phenomenon that the connected probability of a sample determines the popularity of it. Inspired by these observations, we assume that in feature representation learning, the modulus of the learned feature vector is larger if the popularity is greater. So, the two modulus of learned event instances feature vectors of positive-pair will be closer.

For discriminate feature learning of our GCN model, we utilize the popularity of the output event instance feature vector in Z to distinguish different classes. As shown in the Figure 2, for any two learned event instance vectors V_{e_i} and V_{e_j} that satisfy $|V_{e_i}| \geq |V_{e_j}|$, we employ a ratio of modulus $x = \frac{|V_{e_i}|}{|V_{e_j}|}$ as the input of a nonlinear mapping function $f(x) = -log(x - 1 + c)$, where the coefficient c is 0.01 to avoid no upper bound output. We assume that the ratio of modulus of *positive-pair* will belong to $[1, 2)$. So, the nonlinear mapping function $f(x)$ can map the above ratio x from $[1, 2)$ to $(0, 2]$, and $[2, +\infty)$ to $(-\infty, 0)$. Next, we add a Sigmoid function to map the output of the nonlinear mapping layer to 0 or 1 by a threshold 0.5. As shown in the Figure 2, one *positive-pair* or *negative-pair* input sample can only be paired with an output of 0 or 1. For one batch (64 pairs) samples, our model can generate one batch size (1×64) of one-zero output vector. So, we can use a cross entropy function as our model's loss function, and employ the popular stochastic gradient descent (SGD) method to iterate all parameters. The learned weights $\vec{\omega}$ will be used to measure similarity for any two social event instances. To verify the avoidance of over-fitting ability of our model, we can perform over 7000 epochs, and observe that the evaluation criteria of the model changes over time in Section 4.

For the testing of any event instance t from the test set of the original N samples, we first assume that there are a total of C event classes in the original N event instances. Secondly, we calculate the ratio of the modulus of the representation vectors for t and the remaining $N - 1$ samples,

respectively. Then, for each event class, we can get a probability that event instance t most likely belongs to it. If all of the ratios of modulus are 0, the sample itself is a separate event class. Finally, we select the event class with the highest probability as the test output for the event instance t. Note the fact that the event category of the test set may not be included in the event category of the training set. After the previous analysis, we can calculate a similarity for any two event instances under the event-HIN and the weights $\vec{\omega}$. Since our meta-paths have better interpretability, we also can implement a semi-supervised and fine-grained event clustering based on the learned weights of meta-paths and distance-based clustering models.

4 Experiments

In this section, we evaluate the proposed PP-GCN model and similarity measure KIES using real surveillance data collected in two enterprise systems.

Datasets	Train	Validation	Test	Class
Tencent	17,438	5,813	5,812	9,941
Weibo	6,000	2,000	2,000	5,470

Table 1: Description of evaluation datasets.

4.1 Datasets and Settings

We select two independent social media platforms, news APP from Tencent (a popular APP for young people) and Sina Weibo (a hybrid of Twitter and Facebook, the Twitter of China and Chinese Social Media), to collect datasets. Each event instance is a non-repeating social message text. One event is a set of event instances that contain semantically identical information revolving around a real world incident. An event always has a specific time of occurrence. It may involve a group of social users, organizations, participating persons, one or several locations, other types of entities, keywords, topics, etc. In our work, social events cover a wide variety of types, including a large number of events that occur in the real world and spread on social networks, such as earthquakes, national policies, economic crises, and so on. Each event class refers to a unique event. For example, social media's tweet about *Tiger Woods winning the 2019 Masters of Golf* is an influential event in the real world and unlike *Patrick Reid's 2018 Masters of Golf*. These are two different events that happen in the real world and belong to different event categories. The event labels for the Weibo and Tencent datasets are labeled by the outsourcing companies. Both entities and keywords have been manually extracted for the Tencent dataset. Note that the anonymized social users and their friend relationships involved in these two datasets are granted by the two companies for scientific research purposes only. For both of the two datasets, we use 60% of samples as training set, 20% of samples as development set and the remaining 20% of as test set. The statistics of the two datasets is shown in Table 1. We can see that the total number of class is large and the number of samples in per class is small.

We conduct the experiments on event detection and event clustering on these two datasets. The operating system and software platforms are Ubuntu 5.4.0, Tensorflow-gpu (1.4.0) and Python 2.7. The metrics used to evaluate the performance of event detection are the accuracy and F1 score. The metric used to evaluate the performance of event clustering is the normalized mutual information (NMI).

4.2 Baseline Methods

Since the work of fine-grained social event categorization is relatively small, we briefly describe the baseline methods of text matching and text distance. For all the baselines, we use the implementations or open source codes of these models released by authors and other researchers, and report the best performance of the results.

Support Vector Machine with TF-IDF feature (SVM): Support Vector Machine with pair document TF-IDF features is the most classical approach for classification task. We extract the TF-IDF features for social messages, and then use the SVM classifier to implement the multi-class event classification. **Convolutional Matching Architecture-I (ARC-I)** [Hu *et al.*, 2014]: It encodes text pairs by CNNs, and compares the encoded representations of each text with a MLP. **Convolutional Matching Architecture-II (ARC-II)** [Hu *et al.*, 2014]: It builds directly on the interaction space between two texts, and models all the possible combinations of them with 1-D and 2D convolutions. **Match by Local and Distributed Representations (DUET)** [Mitra *et al.*, 2017]: It matches two texts using both local representation and learned distributed representation. **Multiple Positional Semantic Matching (MV-LSTM)** [Wan *et al.*, 2016]: It matches two texts with multiple positional text representations, and aggregates interactions between different positional representations. **Convolutional Deep Structured Semantic Models (C-DSSM)** [Shen *et al.*, 2014]: It learns low-dimensional semantic vectors for input text by CNNs. **Deep Structured Semantic Model (DSSM)** [Huang *et al.*, 2013]: It utilizes a deep neural network to map high-dimensional sparse features into low-dimensional features, and calculates the semantic similarity of the document pair. **Siamese Encoded Graph Convolutional Network (SE-GCN)** [Liu *et al.*, 2018]: It learns vertex representations through a Siamese neural network and aggregates the vertex features though GCNs to generate the document matching.

Term Frequency-Inverse Document Frequency (TF-IDF): It uses the bag-of-words representation divided by each word's document frequency. **Latent Dirichlet Allocation (LDA)** [Blei *et al.*, 2003]: is a celebrated generative model for text documents that learns representations for documents as distributions over word topics. **Marginalized Stacked Denoising Autoencoder (mSDA)** [Chen *et al.*, 2012]: It is a representation learned from stacked denoting autoencoders. **Componential Counting Grid (CCG)** [Perina *et al.*, 2013]: It is a generative model that models documents as a mixture of word distributions and LDA. **Word Move Distance (WMD)** [Kusner *et al.*, 2015]: It measures the dissimilarity between two documents as the minimum amount of distance that words of one document need to travel to reach words of another document. **Knowledge-driven document similarity**

Algorithms	Tencent		Weibo	
	Accuracy	F1	Accuracy	F1
ARC-I	0.5384	0.4868	0.4910	0.4857
ARC-II	0.5437	0.3677	0.5277	0.5137
DUET	0.5625	0.5237	0.5397	0.5523
DSSM	0.5808	0.6468	0.5765	0.5411
C-DSSM	0.6017	0.4857	0.6170	0.5814
MV-LSTM	0.5562	0.6383	0.6252	0.6613
SVM	0.7581	0.7361	0.6511	0.6268
SE-GCN	0.7901	0.7893	0.7063	0.7015
PP-SE-GCN	0.8319	0.8383	0.7317	0.7384
PA-GCN	0.8818	0.8801	0.7567	0.7591
PP-GCN	**0.9252**	**0.9231**	**0.8000**	**0.8134**

Table 2: Accuracy and F1 results of event detection.

measure (KnowSim) [Wang *et al.*, 2016]: It's also a meta-paths instances based document similarity, and hasn't considered the impacts of social users. The weights of meta-paths are estimated by the Laplacian scores of documents.

4.3 Performance Analysis

Table 2 shows the accuracy and F1-score of different algorithms on the task of event detection in Tencent and Weibo datasets. Overall, the proposed PP-GCN model consistently and significantly outperforms all baselines in terms of accuracy and F1. In the Tencent dataset, PP-GCN achieves 13%–56% improvements in terms of accuracy and F1 over all baselines. In the Weibo dataset, PP-GCN achieves 10%–33% improvements in terms of accuracy and F1 over all baselines.

The improvements can be attributed to the three characteristics of proposed models. First, the knowledgeable HIN is better modeling social events than traditional text modeling methods, such as bag-of-words (SVM), N-gram (ARC-I, ARC-II and C-DSSM) and sequence-of-words (MV-LSTM). Our PP-GCN has improved overall by more than 10% in the event detection over the SE-GCN model incorporating structural and conceptual semantics. Second, the combination of KIES based weighted adjacent matrix and Doc2Vec is better for fine-grained event instance representation learning than for feature extraction on text pairs, such as DUET. Third, the classifier based on the ratio of modulus of generated representations of event instances is better than the traditional pairwise distances. Here, we replace the regression module of SE-GCN model by our proposed popularity based classifier, named by PP-SE-GCN, and the performances can be improved 3%-5% in Tencent and Weibo. The 10%-14% improvements from the SE-GCN to the PP-GCN demonstrate the advantages of knowledgeable HIN modeling and the pairwise popularity based feature learning framework.

Furthermore, our PP-GCN model can avoid over-fitting in training. We replace our classifier in PP-GCN by the angle of generated event instances feature vectors based classifier, namely PA-GCN. In Figure 3, we visualize the test accuracies of the PP-GAN and PA-GCN in Tencent and Weibo in 7000 epochs. From Figure 3, we observe that the overall trend of the accuracies of the PP-GCN model is continuously increas-

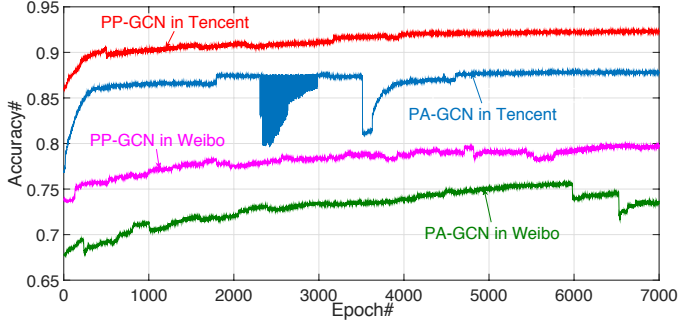| Datasets	TF-IDF	LDA	mSDA	CCG	WMD	KnowSim	KIES	KIES(T)
Tencent	0.6686	0.6979	0.7545	0.7715	0.8166	0.8059	**0.9012**	0.8937
Weibo	0.5824	0.6014	0.6518	0.6973	0.7261	0.7191	0.7820	**0.8041**

Table 3: NMI results of event clustering.

Figure 3: Illustration of the Accuracy for PP-GCN and PA-GCN.

ing, but the accuracies of the angle-based PA-GCN model have periodic fluctuations. Essentially, the ratio of the modulus between vectors is more stable than the angle in iterations. Compared to the angle-based classifier, the popularity-based classifier has better ability to learn discriminating and stable event instance feature and prevent overfitting.

One advantage of the proposed PP-GCN compared to other methods is that the weights $\vec{\omega}$ between the meta-paths can be learned according to the event detection task. Due to the interpretability of the meta-path and similarity measure KIES, the learned weights $\vec{\omega}$ can be utilized in other applications. Here, we make use of different similarity measures including KIES and other methods discussed in Section 4.2, and leverage the popular k-means algorithm to cluster the events. For the KIES distance metric, we use the two weights $\vec{\omega}$ learned in the event detection tasks of Tencent and Weibo datasets, and then calculate the KIES distances between event instances by Eq. 1 to implement a semi-supervised fine-grained event clustering. Note that the test data did not participate in the training when learned the meta-path weights $\vec{\omega}$ in PP-GCN.

As shown in Table 3, our proposed similarity measure KIES achieves the best performances on the two clusters tasks in terms of NMI. Moreover, among the baselines, the WMD, mSDA and CGG measures have been verified to achieve state-of-the-art effects in text similarity in [Kusner *et al.*, 2015]. Compared to other similarity measures, our KIES based k-means method achieves 6%–24% improvements in terms of NMI. We even implement a meta-path weights transfer experiment between Tencent dataset and Weibo dataset. We see that the performance of the Weibo dataset are improved more than 2% when employing the weights of the Tencent dataset to the Weibo by the KIES(T) based k-means

method, but not the other way around. We believe the reason is that Tencent dataset has more applicable meta-path weights by training with manually labeled entities and keywords. Based on the learned weights and the interpretable distance metric KIES, we have achieved the best performance of semi-supervised event clustering.

5 Conclusion

In this paper, we propose a knowledgeable HIN based social event modeling framework, and design a novel pairwise popularity GCN model to learn both meta-paths weights and discriminant event instance representation, and achieves fine-grained social event categorization with state-of-the-art performances. By using the proposed PP-GCN model, we are able to overcome the problems of large category size and sparse small number of samples per class and preventing overfitting in our tasks. Experimental results show that our PP-GCN and KIES similarity measure can significantly outperform state-of-the-art baselines methods on two real-world social datasets. In the future, we plan to study the interpretability of the different importance of meta-paths, and extend our framework to other complex parameter leaning and applications.

References

[Aggarwal and Subbian, 2012] Charu C Aggarwal and Karthik Subbian. Event detection in social streams. In *SDM*, 2012.

[Allan, 2012] James Allan. *Topic Detection and Tracking: Event-based Information Organization*. Springer Publishing Company, Incorporated, 2012.

[Angel *et al.*, 2012] Albert Angel, Nikos Sarkas, Nick Koudas, and Divesh Srivastava. Dense subgraph maintenance under streaming edge weight updates for real-time story identification. *PVLDB*, 2012.

[Atefeh and Khreich, 2015] Farzindar Atefeh and Wael Khreich. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164, 2015.

[Auer *et al.*, 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary

Ives. Dbpedia: A nucleus for a web of open data. In *ASWC*, 2007.

[Becker and Gravano, 2011] Hila Becker and Luis Gravano. *Identification and characterization of events in social media*. Columbia University, 2011.

[Blei *et al.*, 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[Blei *et al.*, 2010] David M Blei, Thomas L Griffiths, and Michael I Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *JACM*, 2010.

[Chandola *et al.*, 2009] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 2009.

[Chen *et al.*, 2012] Minmin Chen, WUSTL EDU, and Zhixiang Eddie Xu. Marginalized denoising autoencoders for domain adaptation. In *ICML*, 2012.

[Griffiths *et al.*, 2004] Thomas L Griffiths, Michael I Jordan, Joshua B Tenenbaum, and David M Blei. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, 2004.

[Hu *et al.*, 2014] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *NIPS*, 2014.

[Huang *et al.*, 2013] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using click-through data. In *CIKM*, 2013.

[Ji and Grishman, 2008] Heng Ji and Ralph Grishman. Refining event extraction through cross-document inference. *Proceedings of ACL*, pages 254–262, 2008.

[Kim *et al.*, 2009] Jin-Dong Kim, Tomoko Ohta, Yoshinobu Kano, and Jun'ichi Tsujii. Overview of bionlp'09 shared task on event extraction. In *BioNLP*, 2009.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Kusner *et al.*, 2015] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *ICML*, 2015.

[Le and Mikolov, 2014] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, 2014.

[Liu *et al.*, 2018] Bang Liu, Ting Zhang, Di Niu, Jinghong Lin, Kunfeng Lai, and Yu Xu. Matching long text documents via graph convolutional networks. *arXiv*, 2018.

[Liu *et al.*, 2019] Yaopeng Liu, Hao Peng, Jianxin Li, Yangqiu Song, and Xiong Li. Event detection and evolution in multi-lingual social streams. *Frontiers of Computer Science*, 2019.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[Miller, 1998] George Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[Mitra *et al.*, 2017] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. Learning to match using local and distributed representations of text for web search. In *WWW*, 2017.

[Papadopoulos *et al.*, 2012] Fragkiskos Papadopoulos, Maksim Kitsak, M Ángeles Serrano, Marián Boguná, and Dmitri Krioukov. Popularity versus similarity in growing networks. *Nature*, 489(7417):537, 2012.

[Perina *et al.*, 2013] Alessandro Perina, Nebojsa Jojic, Manuele Bicego, and Andrzej Truski. Documents as multiple overlapping windows into grids of counts. In *NIPS*, 2013.

[Ritter *et al.*, 2012] Alan Ritter, Oren Etzioni, Sam Clark, et al. Open domain event extraction from twitter. In *KDD*, 2012.

[Shao *et al.*, 2017] Minglai Shao, Jianxin Li, Feng Chen, Hongyi Huang, Shuai Zhang, and Xunxun Chen. An efficient approach to event detection and forecasting in dynamic multivariate social media networks. In *WWW*, 2017.

[Shen *et al.*, 2014] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning semantic representations using convolutional neural networks for web search. In *WWW*. ACM, 2014.

[Shi *et al.*, 2017] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and Philip.S Yu. A survey of heterogeneous information network analysis. *TKDE*, pages 17–37, 2017.

[Sun and Han, 2013] Yizhou Sun and Jiawei Han. Mining heterogeneous information networks: a structural analysis approach. *Acm Sigkdd Explorations Newsletter*, 2013.

[Wan *et al.*, 2016] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. A deep architecture for semantic matching with multiple positional sentence representations. In *AAAI*, 2016.

[Wang *et al.*, 2016] Chenguang Wang, Yangqiu Song, Haoran Li, Ming Zhang, and Jiawei Han. Text classification with heterogeneous information network kernels. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[Wang *et al.*, 2018] Chenguang Wang, Yangqiu Song, Haoran Li, Ming Zhang, and Jiawei Han. Unsupervised meta-path selection for text similarity measure based on heterogeneous information networks. *Data Mining and Knowledge Discovery*, 32(6):1735–1767, 2018.

[Xu *et al.*, 2017] Bo Xu, Yong Xu, Jiaqing Liang, Wanyun Cui, and Yanghua Xiao. Cn-dbpedia: A never-ending chinese knowledge extraction system. In *IEA/AIE*, 2017.

[Yu *et al.*, 2017] Weiren Yu, Jianxin Li, Md Zakirul Alam Bhuiyan, Richong Zhang, and Jinpeng Huai. Ring: Real-time emerging anomaly monitoring system over text streams. *IEEE Transactions on Big Data*, 2017.

[Zacks and Tversky, 2001] Jeffrey M Zacks and Barbara Tversky. Event structure in perception and conception. *Psychological bulletin*, 2001.

[Zhang *et al.*, 2014] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *ECCV*, 2014.