SkipCas: Information Diffusion Prediction Model Based on Skip-Gram

Dedong Ren and Yong Liu(✉)

School of Computer Science and Technology, Heilongjiang University, Harbin, China
2201840@s.hlju.edu.cn, liuyong123456@hlju.edu.cn

Abstract. The development of social network platforms such as Twitter and Weibo has accelerated the generation and transmission of information. Predicting the growth size of the information cascade is widely used in the fields of preventing rumor spread, viral marketing, recommendation system and so on. However, most of the existing methods either cannot fully capture the structural representation of the cascade graph, or cannot effectively utilize the dynamic changes of information diffusion, which often leads to poor prediction results. Therefore, in this paper, we propose a novel deep learning model called SkipCas to predict the growth size of the information cascade. First, we use the diffusion path and time effect at each diffusion time in the cascade graph to obtain the dynamic process of the information diffusion. Second, we put the sequence of biased random walk sampling into the skip-gram model to obtain the structural representation of the cascade graph. Finally, we combine the dynamic diffusion process and the structural representation to predict the growth size of the information cascade. Extensive experiments on two real datasets show that our model SkipCas significantly improves the prediction accuracy compared with the state-of-the-art models.

Keywords: Information cascade · Cascade size prediction · Structural information · Random walk

1 Introduction

Online social networking platforms such as Twitter, Weibo and Facebook have become the main sources of information in people's daily life. Being able to accurately predict the size of information diffusion after a certain period has attracted widespread attention in the academic community, which plays a critical role in suppressing rumors information diffusion, improving content recommendation and other many down-stream applications [1,2].

Many approaches have been proposed for predicting information diffusion. It mainly falls into three categories: 1) Feature-based approaches: They mainly focus on identifying and incorporating hand-crafted features for cascade prediction, such as temporal features [3,4], structural features [5,6], and content features [7,8], etc. Their performance depends on extracted features, which are difficult to generalize to new domains. 2) Generative approaches: The popularity

M.-R. Amini et al. (Eds.): ECML PKDD 2022, LNAI 13714, pp. 258–273, 2023.
https://doi.org/10.1007/978-3-031-26390-3_16

of information cascades over time is considered as a dynamic time series fitting problem [9], leading to the development of certain macroscopic distributions or stochastic processes based on various strong assumptions. These approaches rely heavily on the designed self-excited mechanisms and intensity functions [10,11]. This usually has a huge gap with the real world, resulting in poor predictive power. 3) Deep learning-based approaches: In recent years, researchers leverage various deep learning techniques to capture the temporal and sequential processes of information diffusion. For example, DeepCas [12], Topo-LSTM [13], and DeepCon+Str [14] model the network topology for information diffusion prediction; DeepHawkes [15] and RNN-based CRPP [16] model the temporal information for information diffusion prediction.

Despite obvious improvements in modeling cascade diffusion, existing deep learning methods still face several key challenges: 1) The dynamics of information diffusion are not effectively utilized in existing methods. 2) The structural representation of the cascade network are critical for accurately predicting information cascades. However, most methods fail to fully obtain the structural representation, resulting in unsatisfactory prediction results.

To address the above challenges, we propose a novel information cascade prediction model called SkipCas, which attempts to capture the dynamic diffusion process of the information cascade and obtain the structural representation of the cascade network. To capture the dynamic diffusion process, we put the diffusion path at each diffusion time in the cascade graph into GRU to obtain path representations, weight path representations with diffusion time, and then pool all path representations. To obtain the structural representation of the cascade network, we represent the cascade graph as a set of biased random walk paths and fed them into the skip-gram model to obtain node representations, and then pool all node representations. Finally, we integrate the dynamic diffusion process with the structural representation to predict the growth size of the information cascade. Our main contributions can be summarized as follows:

1) We propose a novel deep learning model called SkipCas for information growth size prediction.
2) We encode the diffusion path at each diffusion time in the cascade graph, which can well preserve the dynamic diffusion process of information diffusion.
3) We leverage the skip-gram model to capture the network structure and obtain the structural representation of the cascade graph.
4) Extensive experiments on several real-world cascade datasets show that SkipCas can significantly improve the cascade size prediction performance compared with the state-of-the-art approaches.

2 Related Works

2.1 Cascades Prediction

The existing methods on information cascade prediction fall into the following three categories:

Feature-based approaches extract various hand-crafted features from the original data, usually including information temporal features [3,4], cascade structural features [5,6], content features [7,8] and user features [17], and then predict its popularity through various machine learning models. However, their performance relies heavily on the relevant features extracted by hand, and may not be directly applied when they are not in a specific environment, thus the feature-based approaches are not easy to generalize.

Generative approaches typically treat the growing size of the information cascade as a cumulative stochastic process [18], modeling it as a parametric model and then estimating the parameters for each event by maximizing the probability of the event occurring at the observed time. [19] divided the observed popularity into multiple stages at equal-sized time intervals, modeled them using multiple linear regression and auto-regression, respectively. In addition to the simple regression-based model, they also used different point processes, such as Poisson [20,21] and Hawkes processes [10,22]. However, as mentioned in [1], the Poisson process is too simple to capture the diffusion patterns, and Hawkes usually overestimates their popularity, probably due to their underlying self-excitation mechanism. In contrast, SkipCas enables incorporates both structural and temporal information.

Deep learning-based approaches are inspired by deep neural networks and have achieved significant performance improvements in many applications. DeepCas [12] is the first deep learning-based information cascade prediction model, which learns the representation of cascade graphs in an end-to-end manner. DeepHawkes [15] inherits the high interpretability of the Hawkes process and has the high predictive ability of deep learning methods. CasCN [23] samples the cascade graph as cascade subgraphs and employs a dynamic multi-directional convolutional network to learn the structural information of the cascade graph. VaCas [24] extends the deterministic cascade embedding with random node representation and diffuse uncertainty, enabling more robust cascade prediction. In addition, methods such as CYAN-RNN [25], Topo-LSTM [13], and SNIDSA [26] extract the full path of diffusion from sequential observations of information infections, using recurrent neural networks and attention mechanisms to model information growth and predict diffusion size. However, they lack better learning ability in cascading structural information and dynamics modeling, due to the bias of sampling methods and the inefficiency of local structure embedding.

2.2 Graph Representation

Learning node embeddings in graphs aims to learn low-dimensional latent representations of nodes in the networks, and the learned feature representations can be used as features for various graph-based tasks, such as classification, clustering, link prediction, and visualization. Word2vec [27] is an unsupervised learning technique that given a word can guess its surrounding context. Inspired by it, the DeepWalk [28] algorithm first introduced a word vector training model to the network. To capture the diversity of network structures, node2vec [29] generated biased second-order random walk, rather than uniform ones. In addition,

inspired by Convolutional Neural Networks, GCN [30] has also been developed to learn representations of nodes in graphs from neighboring node representations, such as GraphSage [31] and DiffPool [32]. We fuse dynamic diffusion processes to predict the cascade growth size based on the skip-gram model.

3 Preliminaries

In this section, we will formally define the cascade prediction problem.

Definition 1. Social Graph. Given a snapshot of a social network graph $G = (V, E)$, where V is the set of vertices of the social graph and $E \subset V \times V$ is the set of edges. A vertex can be a user of a social platform or a paper in the network of academic papers, and an edge represents the relationship between two nodes, such as retweeting or citing.

Definition 2. Cascade Graph. Suppose there are M messages in the social network, for the i-th message we use the cascade graph C_i to represent. Each cascade graph C_i corresponds to an evolution sequence, we use the cascade $g_i(t_j) = \left\{V_i^{t_j}, E_i^{t_j}, t_j\right\}$ to represent the diffusion process of the cascade graph C_i within time t_j, where $V_i^{t_j}$ denotes the users participating in the cascade within time t_j, $E_i^{t_j}$ denotes the feedback relationship between users in $V_i^{t_j}$ (e.g., retweeting or citation), t_j is the time between retweets of the original post. The diffusion process of the cascade graph is shown in Fig. 1, i.e., $g_i(t_0) = \{\{A\}, \{\oslash\}, t_0\}$, $g_i(t_1) = \{\{A, B\}, \{(A, B)\}, t_1\}$, ... , and so on.

Definition 3. Growth Size. In this paper, the growth size of the cascade is defined as the number of retweets or citations of a message or paper. Specifically, given a cascade C_i, within the observation time window T, our research task is to predict the growth size $\triangle S_i$ of C_i at the fixed time interval $\triangle t$, e.g., $\triangle S_i = |V_i^{T+\triangle t}| - |V_i^T|$.

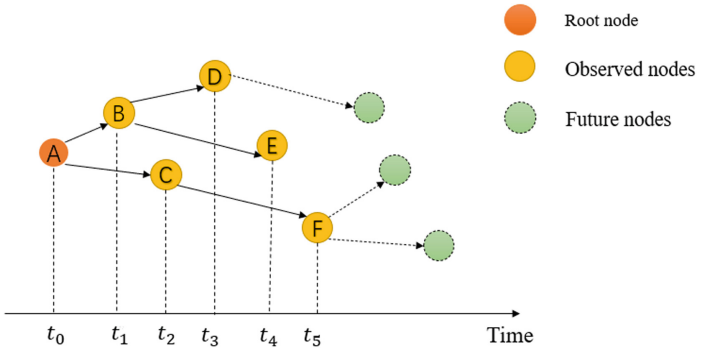

Fig. 1. Diffusion cascade graph of a certain message.

4 Model

The framework of our proposed SkipCas model takes the cascade as input and predicts the growth size $\triangle S_i$ of the cascade graph C_i as output. The model is shown in Fig. 2. SkipCas consists of four main components: 1) Diffusion path coding: the diffusion paths are coded by recurrent neural networks according to the observed cascade diffusion order; 2) Time effect: the encoded diffusion paths combine with temporal effects to further extract the cascade representation; 3) Structural modeling: the sequence of random walk sampling is used to obtain the structural representation of the cascade graph through the skip-gram; 4) Prediction: the cascaded representation with time effect and the structural representation are fed into the multilayer perceptron for cascade size prediction.

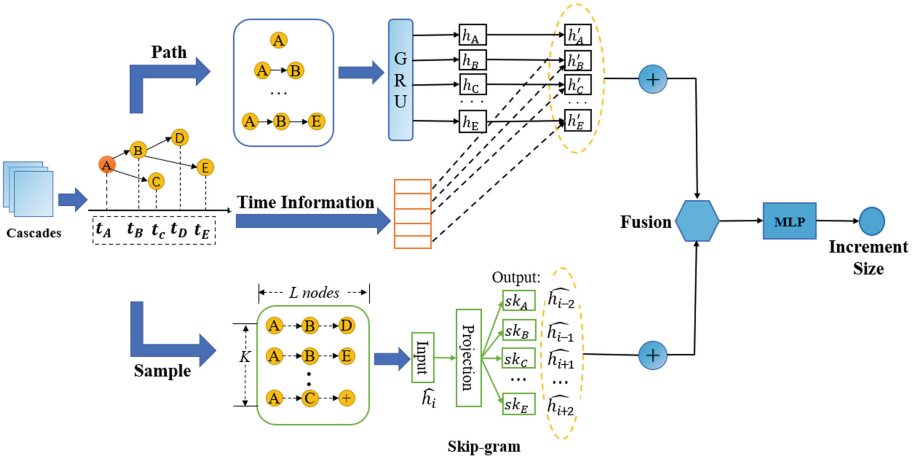

Fig. 2. Framework of SkipCas model.

4.1 Diffusion Path Encoding

Users participating in cascading diffusion will not only be affected by users who have just occurred retweeting behavior, but also by previous users; similarly, previous participants will also influence their direct retweeters and indirect retweeters. As shown in Fig. 1, user A published a message, user B retweeted the message from user A, and D retweeted the message of user B, then the retweet path of this message is A → B → D, user A still has influence on the delivery of the message. This illustrates that each user in the cascade may have an impact on the whole information transfer that follows it. Therefore, we encode the entire cascaded diffusion path.

We use the Gated Recursive Unit (GRU) to encode the entire diffusion path. Specifically, each user in the diffusion path is first represented by a one-hot vector, and then according to the order of the diffusion path, the k-th in the

diffusion path, denoted as $x_k \in R^d$, is fed to the GRU unit. The hidden state $h_k = GRU(x_k, h_{k-1})$ is updated after the update operation on it, where the output $h_k \in R^H$, the input $x_k \in R^d$, h_{k-1} represents the hidden state before the update, d is the dimension size of the user, and H is the dimension size of the hidden state. The update formula of GRU is as follows:

The reset gate $r_k \in R^H$ is calculated by

$$r_k = \sigma(W_r x_k + U_r h_{k-1} + b_r). \tag{1}$$

The update gate $z_k \in R^H$ is calculated by

$$z_k = \sigma(W_z x_k + U_z h_{k-1} + b_z). \tag{2}$$

The actual activation of hidden state h_k is calculated by

$$h_k = z_k \cdot h_{k-1} + (1 - z_k) \cdot tanh(W_h x_k + U_h h_{k-1} + b_h), \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid activation function, $W_r \in R^{H \times d}$, $W_z \in R^{H \times d}$, $W_h \in R^{H \times d}$, $U_r \in R^{H \times H}$, $U_h \in R^{H \times H}$, $U_z \in R^{H \times H}$ and $b_r \in R^H$, $b_z \in R^H$, $b_h \in R^H$ are independent trainable parameters.

4.2 Time Effect

The time effect is a common phenomenon of cascading information diffusion and plays an important role in cascading prediction. For example, a post on Weibo is usually frequently retweeted in the first period after it is published, and the number of retweets decreases with time.

Suppose a cascade C_i whose duration after generation is t, then it is easy to know how long the time interval between its generation and each retweet or citation. Then we can get the time interval of each user's retweet in the cascade graph, e.g., $\left\{ t'_v = t^r_v - t_0 \mid 0 \le t'_v \le t, v \in V_i^{t_j} \right\}$, where t^r_v is the time when user v retweets the message, and t_0 is the original posting time of the post.

In order to learn the effect of time on the cascade, we employ the following time decay effect. Supposing the time window of the observed cascade is [0,T], we divide the time window into l equal-sized time intervals as $\{[t_0, t_1), ..., (t_{l-1}, t_l]\}$, where $t_0 = 0$, $t_l = T$. It can assign a corresponding interval to each diffusion time, thus we can compute the corresponding time interval β of the time decay effect for a retweet at time t:

$$\beta = \lfloor \frac{t'_v}{T/l} \rfloor \quad (0 \le t'_v \le t). \tag{4}$$

The function of the time decay effect is:

$$\lambda_\beta = \frac{1}{1 + \frac{t'_v}{t_0}}. \tag{5}$$

Then we add the time decay effect to the obtained cascaded hidden state h_t, and further obtain

$$h'_t = \lambda_\beta h_t. \tag{6}$$

Summation to obtain the representation vector for the cascade C_i:

$$h'(C_i) = \sum_{t=1}^{T} h'_t. \tag{7}$$

4.3 Structural Modeling

The future size of the cascade depends heavily on who is the information "propagator", i.e., the nodes in the current cascade graph. Therefore, a straightforward way to represent a graph is to treat it as a bag of nodes. However, this approach ignores the structural information in the cascade graph, which is important in predicting diffusion. The biased random walk considers the breadth-first and depth-first sampling strategies, which can better capture the structural information of the cascade graph. Therefore, we represent the cascade graph C_i as a set of cascade paths sampled through multiple biased random walk processes. For each random walk process, we first sample the starting node with the following probability:

$$p(u) = \frac{deg_{C_i}(u) + \alpha}{\sum_{u \in V_{C_i}} (deg_G(u) + \alpha)}, \tag{8}$$

where α is the smoother, deg_{C_i} is the out-degree of node u in cascade C_i, and $deg_G(u)$ is the degree of u in the global graph G, V_{C_i} is the set of nodes in cascaded C_i. Then, after the starting node, the neighboring nodes are sampled with the following probability:

$$p(u \in N_{C_i}(v) \mid v) = \frac{deg_{C_i}(u) + \alpha}{\sum_{u \in N_{C_i}(v)} (deg_G(u) + \alpha)}, \tag{9}$$

where $N_{C_i}(v)$ represents the set of neighbors of v in the cascade graph C_i.

The number and length of random walk sampling sequences play a key role in determining the representation of the cascade graph. Therefore, in order to better perform the sampling process, we set two parameters L and K, where K represents the number of sequences sampled, and L represents the length of each sequence. We fix L and K as constants, the specific settings will be explained in the next section of the experiment. Sampling of a sequence stops when we reach a predefined length L or when we reach a node without any outgoing neighbors. If the length of the one sequence is less than L, the sequence is filled with a special node '+'. This process of sampling sequences continues until K sequences are sampled.

The skip-gram model was originally proposed in [28] and has been applied to deal with word representations in natural language. It aims to classify as many words as possible based on another word in the same sentence. Specifically, the representation of each given word is the input, and the model uses logistic

regression to predict the words within a certain distance before and after the input word in the sentence. Similarly, we use the sequence of nodes obtained by random walk as input, and after the logarithmic function mapping of the projection layer, we get the embedding vector of each node. Suppose $N_{C_i}(v)$ is the neighborhood list of node v generated by the neighborhood sampling strategy, and the embedding representation is denoted as $\hat{H} = \left\{\hat{h}_1, \hat{h}_2, ..., \hat{h}_n\right\}$, where n is the number of nodes. The following objective can be optimized by the skip-gram model to maximize the log-probability of the node neighborhood $N_{C_i}(v)$ for all $v \in V_{C_i}$ as follows:

$$\max_{\hat{H}} \prod_{v \in V_{C_i}} P(N_{C_i}(v) \mid \hat{h}_v). \tag{10}$$

According to the conditional independence assumption, we get:

$$P(N_{C_i}(v) \mid \hat{h}_v) = \prod_{p \in N_{C_i}(v)} P(\hat{h}_p \mid \hat{h}_v). \tag{11}$$

According to the feature space symmetry assumption in Node2vec. We assume that the source node and the neighbor nodes have symmetric effects with each other in the embedding space, the conditional likelihood function for each source-neighbor node pair can be modeled using a softmax function parameterized by the dot product of its features:

$$P(\hat{h}_q \mid \hat{h}_v) = \frac{exp(\hat{h}_p \cdot \hat{h}_v)}{\sum_{q \in V_{C_i}} exp(\hat{h}_q \cdot \hat{h}_v)}. \tag{12}$$

With the above assumptions, the final objective function can be simplified to:

$$\min_{\hat{H}} S^{loss} = \sum_{v \in V_{C_i}} (\log \sum_{q \in V_{C_i}} exp(\hat{h}_q \cdot \hat{h}_v) - \sum_{p \in N_{C_i}(v)} (\hat{h}_p \cdot \hat{h}_v)). \tag{13}$$

4.4 Prediction

We integrate the minimization of the squared loss between the predicted growth size and the ground truth, where a multilayer perceptron is used as the prediction, the formula is as follows:

$$\min_{\theta} O^{loss} = \sum_{i=1}^{M} (\log \Delta S_i - \log \Delta \tilde{S}_i)^2. \tag{14}$$

$$\Delta S_i = MLP(h'(C_i) \oplus \sum_{v \in V_{C_i}} \hat{h}_v). \tag{15}$$

where θ denotes the trainable parameters of the MLP, ΔS_i denotes the predicted growth size for cascade C_i, and $\Delta \tilde{S}_i$ denotes the ground truth.

5 Experiments

In this section, we describe the details of the experiments performed on real-world datasets and the analysis of the results between our proposed model and baseline methods.

5.1 Datasets

We evaluate the effectiveness of the proposed model in two information cascade prediction scenarios and compare it with previous work using publicly available datasets, i.e., Weibo and APS. The statistics of the dataset are shown in Table 1.

Sina Weibo is a public dataset provided by [15], where each tweet and its retweets can form a retweet cascade. We follow a similar experimental setup to [15] with observation time windows of length $T = 1\,\text{h}$, $2\,\text{h}$ and $3\,\text{h}$. Due to the effect of circadian rhythms, we focus on tweets posted between 8 am and 6 pm. We randomly select 70% for training, 15% for validation, and the remaining 15% for testing.

American Physical Society (APS) [20] contains scientific papers published by APS journals. Each paper and its citations in the APS dataset form a citation cascade, and the growth size of the cascade is the number of citations. We only use papers published between 1893 and 1989, so that each paper has at least 20 years to develop its cascade. For the length T of the observation time window, we choose T = 5 years, 7 years and 9 years. Similarly, the first 70% of the data is used for training, 15% for validation, and 15% for testing.

Table 1. Statistics of datasets

	Dataset	Weibo			APS		
Number of Cascades	All	119,311			207,685		
Number of Nodes	All	325,380			616,014		
Number of Edges	All	8,466,858			4,710,547		
T		1 h	2 h	3 h	5 years	7 years	9 years
	Trian	25,515	29,515	31,780	16,299	21,171	24,658
Cascades	val	5,386	6,324	6,810	3,582	4,507	5,254
	Test	5,386	6,324	6,810	3,475	4,589	5,279

5.2 Baselines

We compare the proposed model with some state-of-the-art cascade prediction methods, including:

Feature-Based: Recent studies have shown that structural features, temporal features, and other features (e.g., content features) are useful for information cascade prediction. We select several features commonly used in cascade graphs

(e.g., the number of nodes, the number of edges, average degree, edge density) and predicted the size of the cascade through Feature-linear and Feature-Deep.

Node2vec [29]: It is the representative of node embedding methods. We perform random walks on the cascade graph and generate an embedding vector for each node. Then the embeddings of all nodes in the cascade graph are fed into the MLP for prediction.

DeepCas [12]: The first deep learning architecture for information cascade prediction, which represents the cascade graph as a set of random walk paths via random walks, and uses GRU and attention mechanism to model and predict cascade sizes in an end-to-end manner.

Topo-LSTM [13]: It uses a directed acyclic graph as the diffusion topology, the LSTM is used to model the relationship between nodes in the graph. The hidden state and cell of each node at a given time depends on the hidden state and cell of each previous node that was infected before that time instant.

DeepHawkes [15]: It integrates the high predictive power of deep learning into the interpretable factors of the Hawkes process for cascading size prediction. Bridging the gap between predicting and understanding information cascades.

CasCN [23]: It samples the cascade graph as a sequence of sub-cascade graphs, learns the local structure of each sub-cascade by graph convolution, and then captures the evolution of the cascade structure using LSTM.

DeepCon+Str [14]: It learns the embeddings of the cascade as a whole. It first constructs higher-order graphs based on content and structural similarity to learn the low-dimensional representation of each cascade graph, and then makes cascade predictions through a semi-supervised language model.

5.3 Experimental Settings

The models mentioned above involve several hyper-parameters. For example, the L2 coefficient in Feature-linear is chosen to be 0.05. For Feature-deep, the parameters are similar to deep learning-based approaches. For the sampling sequence of the cascade graph, we set $K = 200$ paths and the length of each path $L = 10$. For Node2vec, we follow the work in [29].

For DeepCas, DeepHawkes, Topo-LSTM, CasCN, DeepCon+Str and our model SkipCas all follow the settings of [12], where the user embedding dimension size is 50, the hidden layer of each GRU is 32 units, and the hidden dimensions of the two-layer MLP are 32 and 16, respectively. The learning rate is 0.005, the batch size is set to 32, and the smoother α is set to 0.01.

5.4 Evaluation Metric

Following the existing work, we adopt mean squared log-transformed error (MSLE) to evaluate the accuracy of predictions on the test set, which is widely used in cascaded prediction evaluation. MSLE is defined as:

$$MSLE = \frac{1}{M}\sum_{i=1}^{M}(\log \Delta S_i - \log \Delta \tilde{S}_i)^2, \tag{16}$$

where M is the total number of messages, ΔS_i denotes the predicted growth size for cascade C_i, and $\Delta \tilde{S}_i$ denotes the ground truth.

Table 2. Overall performance comparison of information cascades prediction among different methods.

Datasets	Weibo			APS		
Metric	MSLE					
T	1 h	2 h	3 h	5 years	7 years	9 years
Features-deep	3.682	3.361	3.296	1.593	1.514	1.465
Features-linear	3.501	3.435	3.324	1.582	1.508	1.456
Node2vec	3.795	3.523	3.513	2.278	2.003	1.982
DeepCas	3.649	3.250	3.056	1.629	1.538	1.467
Topo-LSTM	2.772	2.643	2.423	1.511	1.483	1.462
DeepHawkes	2.501	2.384	2.275	1.286	1.236	1.162
CasCN	2.348	2.243	2.066	1.455	1.353	1.222
DeepCon+Str	2.670	2.391	2.377	1.468	1.382	1.327
SkipCas	**2.251**	**2.103**	**1.890**	**1.163**	**1.086**	**1.045**

5.5 Experimental Results

We compare the performance of the proposed model with several baseline methods on the Weibo and APS datasets, and the results are shown in Table 2. Experimental results show that the SkipCas model performs relatively well on information cascade prediction for both datasets. It not only outperforms traditional methods, but also state-of-the-art deep learning methods, with a statistically significant drop in MSLE. We plot the training process of SkipCas on the Weibo and APS datasets as shown in Fig. 3. It can be seen that the SkipCas loss gradually converges to a lower result.

The performance gap between Feature-deep and Feature-linear is very small, and Feature-linear outperforms Feature-deep on the APS dataset. This means that deep learning does not always perform better than traditional prediction methods if there is a representative set of information cascading features. However, the performance of these methods depends heavily on the relevant features extracted by hand, and it is difficult to generalize to other domains.

For the embedding method, Node2vec performs poorly on both datasets. It only uses the nodes in the graph to represent the network and ignores other structural and content information in the cascade.

DeepCas shows better performance than feature-based methods on the Weibo dataset, but it is inferior to feature-based methods on the APS dataset, which

again shows that deep learning methods are not necessarily better than feature-based methods. However, it still performs worse than other deep learning-based methods because it ignores temporal features and topology of cascaded graphs; similarly, Topo-LSTM lacks temporal features and cannot extract enough information from the cascade, so that its performance is slightly worse compared to our model. DeepHawkes does not consider the topological information of the cascade, and its performance depends on the time series modeling ability. Although CasCN utilizes the structure and time information of the cascade network at the same time, its performance is not the best due to its weak ability to learn structural information. DeepCon+Str utilizes the similarity of cascade graph structure and content to obtain the embedding of the whole cascade graph, but it does not consider the time factor, which affects the prediction performance.

Among these baselines, SkipCas has the best performance and achieves good results on both datasets because it fully investigates the dynamic diffusion process and structural representation of information cascades.

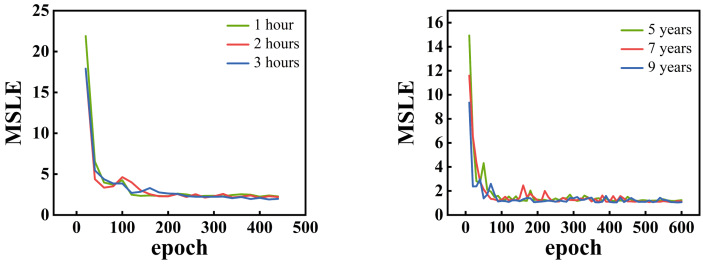

Fig. 3. Convergence of SkipCas on Weibo and APS datasets.

5.6 Ablation Study

To better investigate the effectiveness of each component of SkipCas, we propose four variants. Table 3 summarizes the performance comparison between the models and variants.

SkipCas-LSTM: This method uses LSTM to replace the GRU of the proposed model. Similar to GRU, the LSTM variant models the cascading information through extra gating units.

SkipCas-Time: This method does not consider the time effect of the cascade graph, and is to test the necessity of the time effect in the proposed model.

SkipCas-Path: This method uses a cascade sequence of random walk samples instead of diffusion paths.

SkipCas-Skipgarm: This method does not consider the skip-gram component of the proposed model and only uses GRU and temporal features for prediction, which is to test the importance of the structure of the cascade graph.

From Table 3, we can see that compared with other variants, the prediction error of the original model SkipCas has a certain reduction. Although the error of SkipCas-LSTM is not different from the original model, it can still show that our choice of recurrent neural network is correct; by comparing SkipCas-Time, we find that ignoring the time effect leads to a significant increase in prediction error, which indicates that the time effect is essential in cascading predictions. Similarly, the prediction performance of SkipCas-Path is also decreased significantly, which indicated that the diffusion path could better reflect the change process of the cascade graph. In addition, compared with the original model, the prediction effect of SkipCas-Skipgram is significantly reduced, which fully shows that the structural information of the cascade graph is very important in cascade prediction.

In summary, the time effect of the cascade and the structural information of the cascade are important for future cascade prediction, and our experiments also demonstrate the validity and necessity of the individual components of the proposed model, which essentially improve the performance of the information cascade prediction.

Table 3. Performance comparison between SkipCas and its variants.

Datasets	Weibo			APS		
Metric	MSLE					
T	1 h	2 h	3 h	5 years	7 years	9 years
SkipCas-LSTM	2.301	2.194	1.958	1.325	1.166	1.088
SkipCas-Time	2.523	2.438	2.321	1.582	1.458	1.356
SkipCas-Path	2.332	2.286	2.147	1.465	1.364	1.229
SkipCas-Skipgram	2.495	2.423	2.348	1.529	1.328	1.267
SkipCas	**2.251**	**2.103**	**1.890**	**1.163**	**1.086**	**1.045**

5.7 Parameter Analysis

The observation time window T is an important parameter of the model. As shown in Fig. 4, we can observe that the value of MSLE decreases continuously with increasing observation time on the Weibo dataset, and the prediction error improves by 16% for 3 h compared to 1 h; similarly, the same effect is observed on the APS citation dataset, where the prediction performance continues to improve with the increase of observation years, and the prediction error improves by 10.1% for 9 years compared to 5 years. This shows that as the observation time window T increases, the more information we can observe, the easier it is to make more accurate predictions, which is also a natural result of the increase in training data.

For the time interval l, we choose the datasets with Weibo of 2 h and APS of 7 years for analysis. It can be seen from Fig. 5 (left) that with the increase of

the time interval, the prediction performance of the model gradually improves, but when the time interval exceeds 8, the performance starts to decrease again. Therefore, the experiment in this paper adopts the time interval $l = 8$.

For the user embedding dimension size d, we also choose the datasets with Weibo of 2 h and APS of 7 years. The experimental results are shown in Fig. 5 (right). With the increase of dimension size d, the prediction performance of the model improves. When d is 50, the minimum value of MSLE indicates that the prediction effect is the best at this time. However, when the user dimension size exceeds 50, the prediction performance does not improve but decreases. Therefore, in this paper, the user embedding dimension size d is 50.

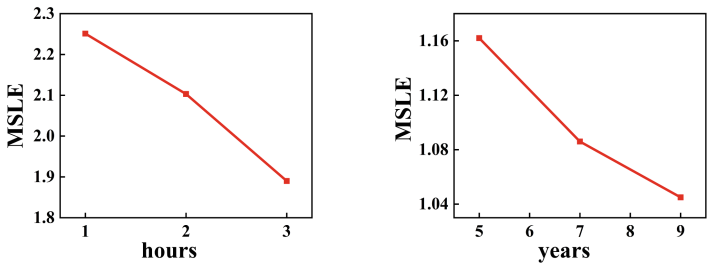

Fig. 4. The effect of observation window on the performance of Weibo (left) and APS (right) datasets.

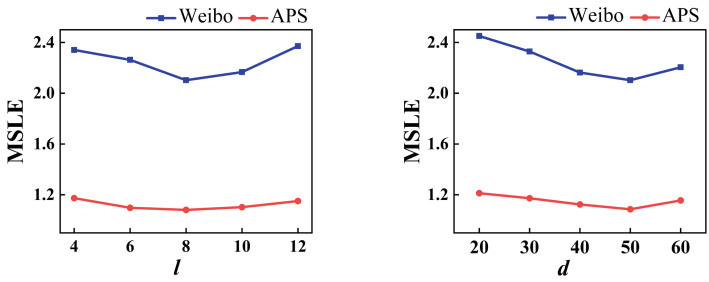

Fig. 5. The effect of time interval l (left) and user embedding dimension size d (right) on datasets performance.

6 Conclusion

In this paper, we propose a novel information cascade prediction model called SkipCas. Our model encodes the diffusion path at each diffusion time in the cascade graph to obtain the dynamic process of information diffusion, uses the sequence of random walk sampling to obtain the structural representation of the cascade graph through skip-gram, and finally predicts the growth size of the information cascade by combining the diffusion process and the structural

representation. The experimental results on two real datasets show that SkipCas significantly improves the cascade prediction performance. As for future works, we plan to incorporate relevant message features such as text content to improve prediction performance and explore more effective methods to further mine the structural information between the cascades.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (No. 61972135), the Natural Science Foundation of Heilongjiang Province in China (No. LH2020F043), and the Foundation of Graduate Innovative Research of Heilongjiang University in China (No. YJSCX2022-236HLJU).

References

1. Gao, X., Cao, Z., Li, S., Yao, B., Chen, G., Tang, S.: Taxonomy and evaluation for microblog popularity prediction. In: TKDD, pp. 1–40 (2019)
2. Zhou, F., Xu, X., Trajcevski, G., Zhang, K.: A Survey of information cascade analysis: models, predictions, and recent advances. ACM Comput Surv. **54**(2), 1–36 (2021)
3. Szabo, G., Huberman, B.A.: Predicting the popularity of online content. Commun. ACM **53**(8), 80–88 (2010)
4. Pinto, H., Almeida, J.M., Gonçalves, M.A.: Using early view patterns to predict the popularity of youtube videos. In: WSDM, pp. 365–374 (2013)
5. Bao, P., Shen, H., Huang, J., Cheng, X.: Popularity prediction in microblogging network: a case study on sina weibo. In: WWW, pp. 177–178 (2013)
6. Weng, L., Menczer, F., Ahn, Y.: Predicting successful memes using network and community structure. In: ICWSM (2014)
7. Tsur, O., Rappoport, A.: What's in a hashtag? Content based prediction of the spread of Ideas in microblogging communities. In: WSDM,, pp. 643–652 (2012)
8. Ma, Z., Sun, A., Cong, G.: On predicting the popularity of newly emerging hashtags in Twitter. Assoc. Inf. Sci. Technol. **64**(7), 1399–1410 (2013)
9. Bao, Z., Liu, Y., Zhang, Z., Liu, H., Cheng, J.: Predicting popularity via a generative model with adaptive peeking window. Phys. A **522**, 54–68 (2019)
10. Zhao, Q., Erdogdu, M.A., He, H.Y., Rajaraman, A., Leskovec, J.: SEISMIC: a self-Exciting point process model for predicting tweet popularity. In: SIGKDD, pp. 1513–1522 (2015)
11. Rizoiu, M., Xie, L., Sanner, S., Cebrian, M., Yu, H., Hentenryck, P.V.: Expecting to be HIP: Hawkes intensity processes for social media popularity. In: WWW, pp. 735–744 (2017)
12. Li, C., Ma, J., Guo, X., Mei, Q.: DeepCas: an end-to-end predictor of information cascades. In: WWW, pp. 577–586 (2017)
13. Wang, J., Zheng, V.W., Liu, Z., Chang, K.C.: Topological recurrent neural network for diffusion prediction. In: ICDM, pp. 475–484 (2017)
14. Feng, X., Zhao, Q., Liu, Z.: Prediction of information cascades via content and structure proximity preserved graph level embedding. Inf. Sci. **560**, 424–440 (2021)
15. Cao, Q., Shen, H., Cen, K., Ouyang, W.R., Cheng, X.: DeepHawkes: bridging the gap between prediction and understanding of information cascades. In: CIKM, pp. 1149–158 (2017)
16. Saha, A., Samanta, B., Ganguly, N., De, A.: CRPP: competing recurrent point process for modeling visibility dynamics in information diffusion. In: CIKM, pp. 537–546 (2018)

17. Cui, P., Jin, S., Yu, L., Wang, F., Zhu, W., Yang, S.: Cascading outbreak prediction in networks: a data-driven approach. In: SIGKDD, pp. 901–909 (2013)
18. Yu, L., Cui, P., Wang, F., Song, C., Yang, S.: From micro to macro: uncovering and predicting information cascading process with behavioral dynamics. In: ICDM, pp. 559–568 (2015)
19. Pinto, H., Almeida, J.M., Gonçalves, M.A.: Using early view patterns to predict the popularity of youtube videos. In: WSDM, pp. 365–374 (2013)
20. Shen, H., Wang, D., Song, C., Barabasi, A.L.: Modeling and predicting popularity dynamics via reinforced poisson processes. In: AAAI, pp. 291–297 (2014)
21. Iwata, T., Shah, A., Ghahramani, Z.: Discovering latent influence in online social activities via shared cascade poisson processes. In: SIGKDD, pp. 266–274 (2013)
22. Zaman, T., Fox, E.B., Bradlow, E.T.: A Bayesian approach for predicting the popularity of tweets. Ann. Appl. Stat. **8**(3), 1583–1611 (2014)
23. Chen, X., Zhou, F., Zhang, K., Trajcevski, G., Zhong, T.: Information diffusion prediction via recurrent cascades convolution. In: ICDE, pp. 770–781 (2019)
24. Zhou, F., Xu, X., Zhang, K., Trajcevski, G., Zhong, T.: Variational information diffusion for probabilistic cascades prediction. In: INFOCOM, pp. 1618–1627, (2020)
25. Wang, Y., Shen, H., Liu, S., Gao, J., Cheng, X.: Cascade dynamics modeling with attention-based recurrent neural network. In: IJCAI, pp. 2985–2991 (2017)
26. Wang, Z., Chen, C., Li, W.: A sequential neural information diffusion model with structure attention. In: CIKM, pp. 1795–1798 (2018)
27. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S.: Distributed representations of words and phrases and their compositionality. In: NIPS, pp. 3111–3119 (2013)
28. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: SIGKDD, pp. 701–710 (2014)
29. Grover, A., Leskovec, J.: Node2vec: scalable feature learning for networks. In: SIGKDD, pp. 855–864 (2016)
30. Kipf, T., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR, (2017)
31. Hamilton, W.L., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NIPS, pp. 1025–1035 (2017)
32. Ying, R., You, J., Morris, C., Ren, X., Hamilton, W.L., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. In: NeurIPS, pp. 4800–4810 (2018)