Information Diffusion Prediction via Cascade-Retrieved In-context Learning

Zhong, Ting; Zhang, Jienan; Cheng, Zhangtao; Zhou, Fan; Chen, Xueqin

DOI
10.1145/3626772.3657909

Publication date
2024

Document Version
Final published version

Published in
SIGIR 2024 - Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval

Citation (APA)
Zhong, T., Zhang, J., Cheng, Z., Zhou, F., & Chen, X. (2024). Information Diffusion Prediction via Cascade-Retrieved In-context Learning. In *SIGIR 2024 - Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2472-2476). Association for Computing Machinery (ACM). https://doi.org/10.1145/3626772.3657909

Information Diffusion Prediction via Cascade-Retrieved In-context Learning

Ting Zhong
zhongting@uestc.edu.cn
University of Electronic Science and Technology of China
Chengdu, Sichuan, China
Kash Institute of Electronics and Information Industry
Kashgar, Xinjiang, China

Jienan Zhang
Zhangtao Cheng*
eroicazjn@outlook.com
zhangtao.cheng@outlook.com
University of Electronic Science and Technology of China
Chengdu, Sichuan, China

Fan Zhou
fan.zhou@uestc.edu.cn
University of Electronic Science and Technology of China
Chengdu, Sichuan, China
Intelligent Terminal Key Laboratory of Sichuan Province
China

Xueqin Chen
X.Chen-10@tudelft.nl
Delft University of Technology,
Faculty of Civil Engineering and Geosciences
Delft, Netherlands

ABSTRACT

Information diffusion prediction, which aims to infer the infected behavior of individual users during information spread, is critical for understanding the dynamics of information propagation and users' influence on online social media. To date, existing methods either focus on capturing limited contextual information from a single cascade, overlooking the potentially complex dependencies across different cascades, or they are committed to improving model performance by using intricate technologies to extract additional features as supplements to user representations, neglecting the drift of model performance across different platforms. To address these limitations, we propose a novel framework called CARE (**CA**scade-**RE**trieved In-Context Learning) inspired by the concept of in-context learning in LLMs. Specifically, CARE first constructs a prompts pool derived from historical cascades, then utilizes ranking-based search engine techniques to retrieve prompts with similar patterns based on the query. Moreover, CARE also introduces two augmentation strategies alongside social relationship enhancement to enrich the input context. Finally, the transformed query-cascade representation from a GPT-type architecture is projected to obtain the prediction. Experiments on real-world datasets from various platforms show that CARE outperforms state-of-the-art baselines in terms of effectiveness and robustness in information diffusion prediction.

CCS CONCEPTS

• **Information systems** → *Information systems applications.*

KEYWORDS

Information diffusion prediction, in-context learning, cascade retrieved.

ACM Reference Format:
Ting Zhong, Jienan Zhang, Zhangtao Cheng, Fan Zhou, and Xueqin Chen. 2024. Information Diffusion Prediction via Cascade-Retrieved In-context Learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3626772.3657909

1 INTRODUCTION

Information diffusion prediction (IDP), also known as *next activated user prediction*, aims to predict the potential users who will be infected based on the observed diffusion cascades and some pertinent knowledge. Accurate predictions can help us understand the dynamics of information spread and users' influence, benefiting various downstream applications, such as recommendation systems [7, 10] and popularity prediction [5, 6, 15, 30, 31, 36]. In recent years, with the advancement and successful application of deep learning techniques in computer vision and natural language processing (NLP), an increasing number of researchers have shifted their focus towards developing deep learning-based methods for IDP. For instance, initially, Topo-LSTM [27], SNIDSA [28], and DeepDiffuse [13] employ Recurrent Neural Networks (RNN [22])-based methods to capture the sequential or topological patterns in historical information diffusion for making predictions. Recently, as Graph Neural Networks (GNNs) [29] have ascended from a niche of representation learning to one of its most coveted methods in various domains, some researchers have begun integrating both RNNs and GNNs to collectively model the comprehensive structural and temporal patterns from information diffusion process [8, 19, 25, 34, 34].
Challenges. In summary, current efforts focus on capturing limited contextual information, such as structural and temporal patterns within individual cascade, or improving user representations through various embedding techniques (e.g., VAE, Hypergraph). While these methods have considerably enhanced accuracy for

*Corresponding author.

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657909

IDP, several key challenges remain unaddressed: (**C1**) Present approaches usually overlook complex interdependencies among cascades, and (**C2**) They do not account for performance variances across different platforms, i.e., meaning that models optimized for a particular social media platform often experience a significant decline on datasets from other platforms, even the diffusion processes are fundamentally similar.

To address these challenges, we propose a novel framework called CARE (**CA**scade-**RE**trieved In-Context Learning), building on the concept of in-context learning (ICL) [3] – a specific method of prompt engineering and has been successfully extended to various tasks beyond large language models (LLMs) [4, 14]. Specifically, CARE first builds a cascade prompts pool derived from all historical cascades in the dataset to capture cross-cascade dependencies. It then uses a search technique to retrieve prompts similar to the query, enabling the model to efficiently leverage relevant past experiences. Moreover, CARE consists of two prompt augmentation strategies and a social relations enhancement embedding module to enrich context, thereby conditioning the model on informative examples to enhance prediction. Finally, CARE utilizes the query-cascade representations generated by a pre-trained GPT architecture [20] for prediction, improving robustness across diverse datasets.

Contribution: Our main contributions are: (1) We propose a novel framework that explores and demonstrates the potential of ICL in modeling the diffusion process. Importantly, CARE does not require the development of a complex model architecture to extract user dependencies or to learn comprehensive representations. (2) We design the dynamic cascade prompt, which differs from the common setting in NLP tasks. Besides, we are among the first to select the most related prompts for effective modeling. (3) We present additional prompt augmentation strategies and a social relations enhancement embedding method that introduce noise into model training and enrich user features. (4) Extensive experiments conducted on two real-world cascade datasets collected from distinct platforms show that our CARE outperforms existing state-of-the-art baselines in terms of effectiveness and robustness.

2 PROBLEM AND METHODOLOGY

Our model architecture is depicted in Figure 1. We focus on learning a model for solving IDP task by incorporating the embedding power of the language foundation model - GPT-type architecture [20], without requiring massive fine-tuning of the backbone. The general definition of IDP is:

Problem Definition: Suppose in a dataset $\mathcal{D}$, we have a collection of historical cascades $C = \{C_1, \cdots, C_{|C|}\}$, and the social graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which encompasses all users present in $\mathcal{D}$. Given a query cascade $\mathbf{q} = \{u_1^q, \cdots, u_m^q\}$, the task of *IDP* aims to learn a model M to predict the next activated user u_{m+1}^q for $\mathbf{q}$ based on C and $\mathcal{G}$. That is, $\hat{u}_{m+1}^q = \mathcal{M}(\mathbf{q}, C; \mathcal{G}; \theta)$, where θ is the model parameters. In the following section, we will reformulate IDP problem w.r.t. ICL, and introduce the details of CARE.

2.1 In-Context Cascade Learning

A common way to format prompts for NLP tasks involves concatenating examples as input-output pairs $(x_i, y_i)_{i=1}^n$, where x_i denotes a question and y_i the corresponding expected response to x_i. IDP is

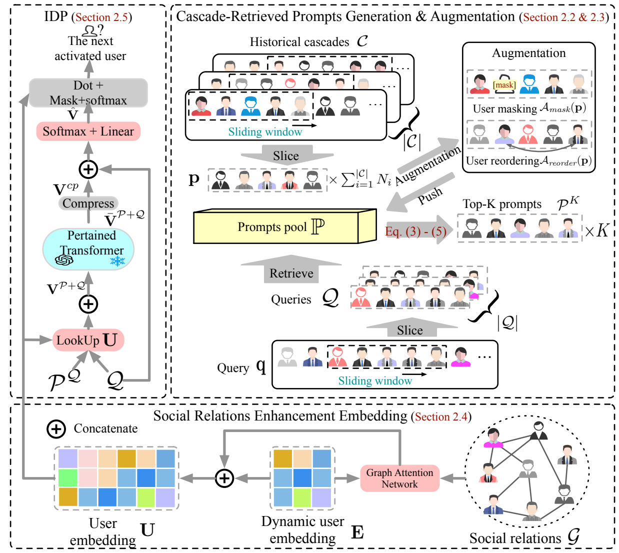

Figure 1: Overview of CARE.

somehow similar to sequential NLP tasks, i.e., each user in the cascade can be analogized to a word in a sentence. Moreover, the diffusion among users forms a dynamic system, where the behavior of one user is influenced by the actions of their preceding user. Consequently, in our work, we coin a new term – dynamic cascade prompt as follows:

Definition 1. *Dynamic cascade prompt (DCP) - In this setting, the prompt is simply the sub-trajectory generated from a given historical diffusion cascade C_i, namely, $\mathbf{p}_i^* = (u_1^*, u_2^*, \cdots, u_{n-1}^*, u_n{}^*)$, where $\{u_i\}_{i=1}^{n-1}$ are inputs (x_i), and $\{u_j\}_{j=2}^{n}$ are outputs (y_i). Notably, the symbol $*$ indicates that multiple prompts can be derived from each historical cascade.*

In CARE, we generate a set of dynamic cascade prompts from the given historical cascades, denoted as $\mathcal{P}$, and then the IDP can be reformulated as: $\hat{u}_{m+1}^q = \mathcal{M}(\mathbf{q}; \mathcal{P}; \mathcal{G}; \theta)$.

2.2 Cascade-Retrieved Prompts Generation

Inspired by document retrieval [18], we design a retrieval-based module to select K most query-relevant cascade prompts from the historical cascades-based prompts pool.

Prompts pool construction: Assuming we have $|C|$ historical cascades C on hand, we first slice each cascade C_i into N_i sub-cascades of fixed length W using a sliding window. All sub-cascades from each cascade are then pushed into a prompts pool $\mathbb{P} = \{\mathbf{p}^j\}_{j=1}^{N_{\mathbb{P}}}$, with a total size of $N_{\mathbb{P}} = \sum_{i=1}^{|\mathbb{C}|} N_i$, and each $\mathbf{p}_j = (u_1^j, \cdots, u_W^j)$.

Query q slicing: Considering the length of the query cascade $|\mathbf{q}|$ may exceed W, we employ a similar operation as used on historical cascades for ease of retrieval, thereby splitting $\mathbf{q}$ into a list of sub-queries, denoted as $\mathcal{Q}$.

Prompts retrieval: Then we employ search engine techniques [18] to retrieve the most related prompts for a given query $\mathbf{q} \in \mathcal{Q}$ [1] from $\mathbb{P}$. Firstly, we employ a Boolean query operation [23] to filter out candidate prompts $\mathcal{P}^{can} = \{\hat{\mathbf{p}}^c\}_{c=1}^{N_{can}}$ from the pool that contains at least one user in common with the query, where N_{can} denotes

[1]Notably, here $\mathbf{q} = (u_1^q, \cdots, u_W^q)$.

the number of prompts in $\mathcal{P}^{can}$. Subsequently, we utilize a ranking function (e.g., BM25 [21]) to calculate the relevant score R for each candidate $\hat{\mathbf{p}}^c$ w.r.t the query q, i.e.,

$$\mathcal{R}(\mathbf{q}, \hat{\mathbf{p}}^c) = \sum_{w=1}^{W} \mathrm{IDF}(u_w^q) \frac{\mathrm{TF}\left(u_w^q, \hat{\mathbf{p}}^c\right) \cdot (k_1 + 1)}{TF\left(u_w^q, \hat{\mathbf{p}}^c\right) + k_1} \tag{1}$$

where k_1 and b are free parameters. $\mathrm{TF}(u_w^q, \hat{\mathbf{p}}^c)$ represents the term frequency of user u_w^q in $\hat{\mathbf{p}}^c$, specifically, it was calculated via Jaccard similarity:

$$\mathrm{TF}\left(u_w^q, \hat{\mathbf{p}}^c\right) = \frac{\left|u_w^q \cap \hat{u}_w^c\right|}{\left|u_w^q \cup \hat{u}_w^c\right|}, \hat{u}_w^c \in \hat{\mathbf{p}}^c, \tag{2}$$

IDF can be regarded as a penalizing factor, which diminishes the importance of common users relative to rare users, implies a match with a rare user yields a stronger similarity signal compared to a commonly seen user. And IDF term is defined as:

$$\mathrm{IDF}(u_w^q) = \log \frac{N_{can} - \mathcal{N}(u_w^q) + 0.5}{\mathcal{N}(u_w^q) + 0.5}, \tag{3}$$

where $\mathcal{N}(u_w^q)$ represents a function used to statistic the number of candidate prompts in which the user u_w^q appears. After ranking the candidate prompts using BM25, we then select the Top-K prompts $\mathcal{P}^K = \{\mathbf{p}^{r_1}, \ldots, \mathbf{p}^{r_K}\}$ as the final ICL inputs.

2.3 Cascade Prompt Augmentation

In CARE, we also design two augmentation strategies to introduce noise into the prompts, aiming for effective model training. Specifically, these strategies are applied following the construction of the prompt pool:

User Masking: Inspired by "word dropout", which is widely adopted to avoid over-fitting in many NLP tasks [2, 9, 11]. In this work, for each $\mathbf{p} = (u_1, \cdots, u_w) \in \mathbb{P}$, we randomly mask $\lfloor \gamma * \mathcal{W} \rfloor$ users with a masking rate γ. Each masked element is replaced by a special token "[mask]", and the formed user masking augmented prompt is denoted as $\mathbf{p}^{mask} = \mathcal{A}_{mask}(\mathbf{p})$.

User Reordering: Existing methods for IDP ground in an assumption that adjacent users in cascades are sequentially dependent [13, 28]. However, in reality, the order of users' behaviors could be flexible due to various unobservable external factors [1, 16]. To reduce the model's dependency on the order of users and to enhance its robustness against new interactions, we design a user reordering method. Specifically, we alter the order of a continuous subsequence of users in $\mathbf{p}$ with a length of $\lfloor \beta * \mathcal{W} \rfloor$ by randomly shuffling their positions. Here, β is the reordering rate. The user reordering augmented prompt can be represented as: $\mathbf{p}^{reorder} = \mathcal{A}_{reorder}(\mathbf{p})$. And the final augmented prompts pool are the collection of masked prompts and reordered prompts, i.e., $\mathbb{P} = \{\mathcal{P}^{mask}, \mathcal{P}^{reorder}\}$.

2.4 Social Relations Enhancement

Similar to the settings in NLP, where the inputs of the transformer backbone are word embeddings, we utilize user dynamic embedding with social relations enhancement in this work to generate user embeddings.

Dynamic user embedding: Suppose we have a trainable embedding matrix $\mathbf{E} \in \mathbb{R}^{N \times d_E}$, where N is the total number of users in a dataset, and d_E is an adjustable dimension.

Social relations enhancement: Since the infected behavior among users is highly influenced by their interrelationships. We then extract social relations from a static social network as a supplementary to the user embedding. Specifically, we use a multi-layer Graph Attention Network (GAT [26]) equipped with multi-head attention to encode the social network graph. And the obtained social relation embeddings for users are $\mathbf{U}_s^{(l+1)} = \sigma(\sum_{k=1}^{K} \mathbf{A}^k \mathbf{W}^k \mathbf{U}_s^{(l)}) \in \mathbb{R}^{N \times d_E}$. Here, $\mathbf{W}^k$ is a set of independent trainable weight matrices and K is the number of attention heads. $\mathbf{A}^k$ denotes the attention matrix calculated through self-attention mechanism (refer to [26]), and $\sigma(\cdot)$ represents activation function. The initial input of GAT is $\mathbf{U}_s^0 = \mathbf{E}$.

Subsequently, we concatenate $\mathbf{E}$ with $\mathbf{U}$ to form the final user embedding matrix $\mathbf{U} \in \mathbb{R}^{N \times d_E}$. And $\mathbf{U}$ converts each user into its individual embedding by looking up the user index $\mathbf{u}_i = \mathrm{LookUp}(u_i, \mathbf{U})$.

2.5 Information Diffusion Prediction

After retrieving the prompts for all queries in Q, we obtain $\mathcal{P}^Q$ with $K \times |Q|$ prompts. Next, for each user in $\mathcal{P}^Q$ and Q, we convert their index into embeddings by looking up the user embedding matrix $\mathbf{U}$. Subsequently, we concatenate these embedding-transformed prompts and queries ($\mathbf{V}^{\mathcal{P}+Q} \in \mathbb{R}^{|Q|(K+1) \times d_E}$), feeding them into a frozen GPT-type backbone as illustrated in Figure 1. This yields the output representation $\bar{\mathbf{V}}^{\mathcal{P}+Q} \in \mathbb{R}^{|Q|(K+1) \times d_E}$. For the final prediction, we compress $\bar{\mathbf{V}}^{\mathcal{P}+Q}$ into $\mathbf{V}^{cp}$ with $(|Q|, d_E)$ shape by sum-polling operation, and concatenate it with queries embedding $\mathbf{V}^Q$. Through a linear layer with softmax, we obtain the final representation $\hat{\mathbf{V}} = \mathrm{softmax}((\mathbf{V}^{cp} \oplus \mathbf{V}^Q)\mathbf{W} + \mathbf{b})$. And the probabilities for all users in $\mathbf{q}$ are calculated by $\mathbf{y}^\mathbf{q} = \mathrm{softmax}(\hat{\mathbf{V}}\mathbf{U}^\top + \mathbf{M}_{mask})$, where $\mathbf{M}_{mask}$ is used to mask users who have already been activated. The overall objective is to minimize the cross-entropy loss between ground truths $\mathbf{y}^\mathbf{q}$ and predictions $\hat{\mathbf{y}}^\mathbf{q}$ [8].

3 EXPERIMENTS

3.1 Experimental Settings

Datasets. We conduct experiments on two real-world datasets collected from different platforms, i.e., Twitter [12], and Douban [35]. Detailed statistics are shown in Table.1.

Table 1: Statistics of the datasets.

Dataset	#Users	#Cascades	#Train	#Val	#Test	Avg.Length
Twitter	12,627	3,454	2,763	345	346	38.22
Douban	12,232	3,485	2,788	348	349	23.09

Evaluation Metrics. Similar to the previous work [33, 34], we employ *Hits score on top k* (Hits@k) and *Mean Average Precision on top k* (MAP@k) for model evaluation, k = [10, 50, 100].

Baselines. We compare CARE with the following state-of-the-art baslines: Topo-LSTM [27], NDM [32], Inf-VAE [24], FOREST [33], DyHGCN [34], MS-HGAT [25], RotDiff [17], and DisenIDP [8].

Parameter Settings. Dataset splitting follows [8]. All baselines follow the same settings in the original papers. The maximum length of $\mathbf{q}$ is set to 200. Our CARE is implemented with PyTorch

Table 2: Experimental results on three datasets (%) (Hits@K scores for K = 10,50,100).

Dataset	Twitter			Douban		
Hits@K	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**
Topo-LSTM	10.45	18.89	25.42	8.97	16.33	21.57
NDM	17.88	25.70	29.96	7.28	14.62	19.26
Inf-VAE	14.93	33.52	46.42	10.94	21.02	34.72
FOREST	25.24	37.57	46.39	18.42	28.54	31.63
DyHGCN	27.68	46.49	57.44	15.90	28.71	36.18
MS-HGAT	34.63	47.52	54.29	20.16	33.46	40.34
RotDiff	33.91	50.78	60.60	20.20	34.10	42.82
DisenIDP	34.96	51.14	59.54	19.96	35.16	42.94
CARE	**38.50**	**53.67**	**63.05**	**24.30**	**36.38**	**43.26**

Table 3: Experimental results on three datasets (%) (MAP@K scores for K = 10,50,100).

Dataset	Twitter			Douban		
MAP@K	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**
Topo-LSTM	9.51	13.68	14.68	6.67	7.63	7.88
NDM	12.24	12.50	12.66	3.39	3.72	3.79
Inf-VAE	19.83	20.68	21.82	7.32	7.98	8.03
FOREST	16.81	17.36	17.42	8.41	10.73	10.77
DyHGCN	16.37	17.22	17.25	8.48	9.06	9.16
MS-HGAT	18.81	19.52	19.92	10.24	10.87	10.98
RotDiff	21.88	22.64	22.78	10.36	11.06	11.18
DisenIDP	22.03	22.76	22.87	9.89	10.59	10.70
CARE	**25.28**	**26.02**	**26.15**	**14.24**	**14.80**	**14.90**

and chooses Huggingface GPT2 [2] as the backbone. We adopt Adam as the optimizer, with a learning rate of 0.005. The batch size is 32, and $d_E = 64$. The masking and reordering rates are 0.2 and 0.4, respectively. CARE employs 2 layers of single-head GAT.

3.2 Evaluation Results

Overall Performance. The overall results are shown in Table 2 and Table 3. We can observe that:

(**O1**) Our model CARE consistently outperforms all baselines across all datasets, showcasing its effectiveness and robustness. Specifically, with $k = 10$, CARE achieves more than 10% improvements in all metrics compared to the best baseline. Furthermore, a comparison between RotDiff and DisenIDP reveals a phenomenon of performance variation across different datasets.

(**O2**) Sequential models, i.e., Topo-LSTM and NDM, exhibit the lowest performance due to their inability to account for the dynamic shifts of user influence, as well as the structural patterns implied in cascades.

(**O3**) The remaining GNN-based baselines outperform sequential models, highlighting the benefit of IDP by using GNNs to extract structural patterns and social relations from cascades and social network, respectively. However, their performance significantly falls short of our CARE, as these methods only focus on individual cascades and overlook the complex interdependencies among different cascades.

Ablation Study. We conduct ablation studies to demonstrate the effectiveness of CARE's key components. The results reported in Table 4 reveal that: (1) Without prompts (*w/o Prompt*), model performance significantly declines, which implies that our prompt design is useful and accurate. (2) Augmentation strategies (*w/o Reordering & w/o Masking*) impose noise to prompts indeed help improve model performance. (3) Replacing the GPT backbone with a trainable LSTM (*LSTM*) fails to enhance the quality of query cascade representation. (4) Both dynamic embedding (*w/o* **E**) and social relations (*w/o* $\mathcal{G}$) are crucial for IDP. And (5) GAT is more efficient than vanilla GCN (*GCN*) in learning interrelations among users because of the attention mechanism.

Table 4: Ablation study on key components of CARE.

Model		Twitter Hits@100	Twitter MAP@100	Douban Hits@100	Douban MAP@100
CARE	**All**	**63.05**	**26.15**	**43.26**	**14.90**
Cascades Prompt	w/o Reordering	61.10	24.59	42.73	12.08
	w/o Masking	60.12	24.49	42.12	12.45
	w/o Prompt	57.93	23.98	38.73	10.96
	LSTM	54.96	21.56	38.58	10.81
Enhanced User Embedding	w/o **E**	60.98	24.27	42.54	11.09
	GCN	60.79	24.15	42.25	11.14
	w/o $\mathcal{G}$	59.15	24.29	41.50	10.99

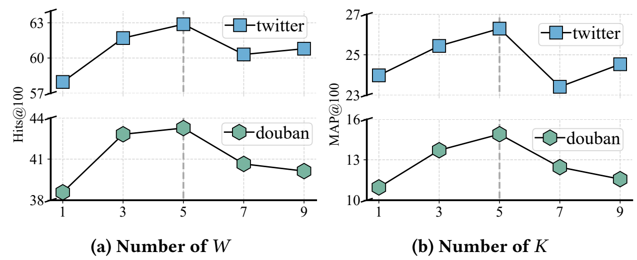

Figure 2: Sensitivity Analysis of CARE on two datasets. We run each model five times and report the mean Hits@100 and MAP@100 respectively.

Sensitivity Analysis. *(1) Sliding window size W for generating the prompt.* Figure 2(a) illustrates the model's performance across various sliding window sizes (choose from $\{1, 3, 5, 7, 9\}$). We can see the optimal size for W appears to be 5. CARE's performance initially improves with increasing sliding window size and subsequently decreases when W becomes larger. *(2) The number of retrieved cascade prompts – K.* Figure 2(b) verifies that using a retrieval operation to select the most relevant prompts significantly enhances model performance, with the best outcomes observed when K increases to 5. However, more prompts ($K > 5$) result in a degradation of performance. We speculate that this is due to the superfluous prompts that induce noise to the model.

4 CONCLUSION

In this paper, we presented a novel in-context learning-based framework – CARE for information diffusion prediction. Specifically, CARE leverages augmented cascade-retrieved prompts alongside a frozen GPT backbone, enhanced by the trainable social relations embedding module and additional linear layers, to achieve accurate predictions. Extensive experimental results on two real-world datasets demonstrate the effectiveness and robustness of CARE.

5 ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No.62176043 and No.62072077) and the Grant SCITLAB-30002 of Intelligent Terminal Key Laboratory of Sichuan Province.

[2]https://huggingface.co/docs/transformers/v4.15.0/model_doc/gpt2

REFERENCES

[1] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised Neural Machine Translation. In *International Conference on Learning Representations (ICLR)*.

[2] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In *Conference on Computational Natural Language Learning (CoNLL)*. 10–21.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 1877–1901.

[4] Mengyang Chen, Lingwei Wei, Han Cao, Wei Zhou, and Songlin Hu. 2023. Can Large Language Models Understand Content and Propagation for Misinformation Detection: An Empirical Study. *arXiv preprint arXiv:2311.12699* (2023).

[5] Xueqin Chen, Fengli Zhang, Fan Zhou, and Marcello Bonsangue. 2022. Multi-scale graph capsule with influence attention for information cascades prediction. *International Journal of Intelligent Systems* 37, 3 (2022), 2584–2611.

[6] Xueqin Chen, Kunpeng Zhang, Fan Zhou, Goce Trajcevski, Ting Zhong, and Fengli Zhang. 2019. Information cascades modeling via deep multi-task learning. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 885–888.

[7] Zhangtao Cheng, Joojo Walker, Ting Zhong, and Fan Zhou. 2022. Modeling multi-view interactions with contrastive graph learning for collaborative filtering. In *International Joint Conference on Neural Networks (IJCNN)*. 1–8.

[8] Zhangtao Cheng, Wenxue Ye, Leyuan Liu, Wenxin Tai, and Fan Zhou. 2023. Enhancing Information Diffusion Prediction with Self-Supervised Disentangled User and Cascade Representations. In *International Conference on Information and Knowledge Management (CIKM)*. 3808–3812.

[9] Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. *Advances in Neural Information Processing Systems (NIPS)* 28 (2015).

[10] Hanwen Du, Huanhuan Yuan, Zhen Huang, Pengpeng Zhao, and Xiaofang Zhou. 2023. Sequential Recommendation with Diffusion Models. *arXiv preprint arXiv:2304.04541* (2023).

[11] Shubhashri G, Unnamalai N, and Kamalika G. 2018. LAWBO: a smart lawyer chatbot. In *ACM India Joint International Conference on Data Science and Management of Data*. Association for Computing Machinery, 348–351.

[12] Nathan O Hodas and Kristina Lerman. 2014. The simple rules of social contagion. *Scientific reports* 4, 1 (2014), 4343.

[13] Mohammad Raihanul Islam, Sathappan Muthiah, Bijaya Adhikari, B Aditya Prakash, and Naren Ramakrishnan. 2018. Deepdiffuse: Predicting the'who'and'when'in cascades. In *IEEE international conference on data mining (ICDM)*. 1055–1060.

[14] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2024. Time-llm: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations (ICLR)*.

[15] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 137–146.

[16] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised Machine Translation Using Monolingual Corpora Only. In *International Conference on Learning Representations (ICLR)*.

[17] Hongliang Qiao, Shanshan Feng, Xutao Li, Huiwei Lin, Han Hu, Wei Wei, and Yunming Ye. 2023. RotDiff: A Hyperbolic Rotation Representation Model for Information Diffusion Prediction. In *International Conference on Information and Knowledge Management (CIKM)*. 2065–2074.

[18] Jiarui Qin, Weinan Zhang, Rong Su, Zhirong Liu, Weiwen Liu, Ruiming Tang, Xiuqiang He, and Yong Yu. 2021. Retrieval & interaction machine for tabular data prediction. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 1379–1389.

[19] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. 2018. DeepInf: Social Influence Prediction with Deep Learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 2110–2119.

[20] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).

[21] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[22] Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. 2017. Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078* (2017).

[23] Gerard Salton. 1986. Another look at automatic text-retrieval systems. *Commun. ACM* 29, 7 (1986), 648–656.

[24] Aravind Sankar, Xinyang Zhang, Adit Krishnan, and Jiawei Han. 2020. Inf-VAE: A variational autoencoder framework to integrate homophily and influence in diffusion prediction. In *International Conference on Web Search and Data Mining (WSDM)*. 510–518.

[25] Ling Sun, Yuan Rao, Xiangbo Zhang, Yuqian Lan, and Shuanghe Yu. 2022. MS-HGAT: memory-enhanced sequential hypergraph attention network for information diffusion prediction. In *AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 36. 4156–4164.

[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations (ICLR)*.

[27] Jia Wang, Vincent W Zheng, Zemin Liu, and Kevin Chen-Chuan Chang. 2017. Topological recurrent neural network for diffusion prediction. In *IEEE international conference on data mining (ICDM)*. 475–484.

[28] Zhitao Wang, Chengyao Chen, and Wenjie Li. 2018. A sequential neural information diffusion model with structure attention. In *International Conference on Information and Knowledge Management (CIKM)*. 1795–1798.

[29] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems (TNLLS)* 32, 1 (2020), 4–24.

[30] Xovee Xu, Fan Zhou, Kunpeng Zhang, and Siyuan Liu. 2022. Ccgl: Contrastive cascade graph learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 35, 5 (2022), 4539–4554.

[31] Xovee Xu, Fan Zhou, Kunpeng Zhang, Siyuan Liu, and Goce Trajcevski. 2021. Casflow: Exploring hierarchical structures and propagation uncertainty for cascade prediction. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 35, 4 (2021), 3484–3499.

[32] Cheng Yang, Maosong Sun, Haoran Liu, Shiyi Han, Zhiyuan Liu, and Huanbo Luan. 2019. Neural diffusion model for microscopic cascade study. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 33, 3 (2019), 1128–1139.

[33] Cheng Yang, Hao Wang, Jian Tang, Chuan Shi, Maosong Sun, Ganqu Cui, and Zhiyuan Liu. 2021. Full-scale information diffusion prediction with reinforced recurrent networks. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)* (2021).

[34] Chunyuan Yuan, Jiacheng Li, Wei Zhou, Yijun Lu, Xiaodan Zhang, and Songlin Hu. 2021. DyHGCN: A dynamic heterogeneous graph convolutional network to learn users' dynamic preferences for information diffusion prediction. In *European Conference on Machine Learning and Knowledge Discovery in Databases (PKDD)*. 347–363.

[35] Erheng Zhong, Wei Fan, Junwei Wang, Lei Xiao, and Yong Li. 2012. Comsoc: adaptive transfer of user behaviors over composite social network. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 696–704.

[36] Fan Zhou, Xovee Xu, Kunpeng Zhang, Goce Trajcevski, and Ting Zhong. 2020. Variational information diffusion for probabilistic cascades prediction. In *IEEE Conference on Computer Communications (INFOCOM)*. 1618–1627.