News Graph: An Enhanced Knowledge Graph for News Recommendation

Danyang Liu
University of Science and Technology of China
Hefei, China
ldy591@mail.ustc.edu.cn

Ting Bai
Beijing University of Posts and Telecommunications
Beijing, China
baiting@ruc.edu.cn

Jianxun Lian
Microsoft Research Asia
Beijing, China
Jianxun.Lian@microsoft.com

Guangzhong Sun
University of Science and Technology of China
Hefei, China
gzsun@ustc.edu.cn

Wayne Xin Zhao
Ji-Rong Wen
Renmin University of China
Beijing, China
batmanfly@gmail.com
jirong.wen@gmail.com

Xing Xie
Microsoft Research Asia
Beijing, China
Xing.Xie@microsoft.com

ABSTRACT

Knowledge graph, which contains rich knowledge facts and well structured relations, is an ideal auxiliary data source for alleviating the data sparsity issue and improving the explainability of recommender systems. However, preliminary studies usually simply leverage a generic knowledge graph which is not specially designed for particular tasks. In this paper, we consider the scenario of news recommendations. We observe that both collaborative relations of entities (*e.g.,* entities frequently appear in same news articles or clicked by same users) and the topic context of news article can be well utilized to construct a more powerful graph for news recommendations. Thus we propose an enhanced knowledge graph called **news graph**. Compared with a generic knowledge graph, the news graph is enhanced from three aspects: (1) adding a new group of entities for recording topic context information; (2) adding collaborative edges between entities based on users' click behaviors and co-occurrence in news articles; and (3) removing news-irrelevant relations. To the best of our knowledge, it is the first time that a domain specific graph is constructed for news recommendations. Extensive experiments on a real-world news reading dataset demonstrate that our news graph can greatly benefit a wide range of news recommendation tasks, including personalized article recommendation, article category classification, article popularity prediction, and local news detection.

CCS CONCEPTS

• **Information systems → Collaborative filtering**; **Web searching and information discovery**; *Data mining*; *Document representation*;

KEYWORDS

News Graph, Collaborative Relations, Recommender Systems, knowledge graph

KaRS 2019, November 3rd-7th, 2019, Beijing, China.

https://doi.org/

ACM Reference format:
Danyang Liu, Ting Bai, Jianxun Lian, Guangzhong Sun, Wayne Xin Zhao, Ji-Rong Wen, and Xing Xie. 2019. News Graph: An Enhanced Knowledge Graph for News Recommendation. In *Proceedings of KaRS 2019 Second Workshop on Knowledge-Aware and Conversational Recommender Systems, Beijing, China., November 3rd-7th, 2019 (KaRS 2019),* 7 pages.
https://doi.org/

1 INTRODUCTION

Due to the explosive growth of information, online news services have become increasingly important for people to get information and understand the outside world. Online news platforms, such as Google News[1] and MSN News[2], contain rich content and contextual information, pertaining to groups of society, politics, entertainment and so on. Although the enormous amount of news streams can be widely applicable to different people preferences, it can also cause information overwhelming to users. Due to the time sensitiveness of news articles, users' interactions with news articles are highly sparse, which results in the data sparsity problem of recommendation systems. To address this challenge, some previous studies such as [12, 16] utilize rich content features in news to model users' preference. Recently, external Knowledge Graph (KG) information, which contains rich knowledge facts and well-structured relations, is also incorporated to alleviate the data sparsity issue and improve the explainability of recommender systems [26]. By using the rich information from an extra KG, the data sparsity problem can be alleviated to some extent.

However, some news specific information is missing in a generic KG, including the collaborative relations of entities encoded in news articles and browsing behaviors of users. Such collaborative relations reveal the context similarity of entities in news and have been rarely explored in news recommendation. For instance, entities that frequently co-occur in articles or clicked by same users are usually strongly related in the news domain. A news article like " *Rihanna* shows support for *LeBron James* in Game 7 vs. Celtics", may indicate a new relation between Rihanna and James that does

[1] https://news.google.com/?hl=en-US&gl=US&ceid=US/
[2] https://www.msn.com/en-us/news

not exist in a generic KG. Also, all of the previous studies overlook the semantic topics of the article itself. We find topics are also important factors to attract users' attention and can benefit to construct a more powerful graph if well utilized. Moreover, as for the information of KG used in recommendation systems, previous graph based studies indiscriminately utilize the KG entities [17, 26], while ignore the fact that some entities and relations in a generic KG may contain irrelevant information for news recommendations. *e.g., The birthday of Donald Trump* is an uninformative relation for news recommendations and including it may even result in the inefficiency problem.

Based on above considerations, in this paper, we propose to utilize the collaborative information from news content and user behaviors to construct a more powerful knowledge graph, named *News Graph* (NG). We first remove the news-irrelevant relations in the original KG, then add two new types of information into NG, *i.e.,* article topic entities and the collaborative edges among KG nodes. For topic entities, we consider both the explicit and implicit topics, *i.e.,* categories of news articles and LDA [15] topics. As for the collaborative edges, we consider the associations among entities from content information of articles and users' reading behaviors. In particular, the collaborative edges are extracted in three ways: (1) co-occurrence in the same news; (2) clicked by the same user; and (3) clicked by the same user in the same browsing session. Due to the selection of news-relevant relations and enhancement of collaborative information, the resulting *news graphs* is expected to possess a stronger capacity of representing news articles and users' reading behaviors, thus it is news domain-oriented. To verify it, we have conduct experiments on four different news recommendation tasks, including personalized item recommendation, news category classification, news popularity prediction and local news detection. Results consistently demonstrate that leveraging our news graph is much more effective than leveraging a generic knowledge graph.

Our contributions are summarized as follows:

- To the best of our knowledge, it is the first time that a domain specific graph, *i.e.,* news graph, is constructed for serving news recommendations. Compared with a generic knowledge graph, we remove the news-irrelevant relations, while add some news topic entities and collaborative relations to make the graph more suitable for news recommendations.
- To construct the news graph, we propose a News Relation Selection (NRS) algorithm to select the news-relevant relations. Meanwhile, we incorporate more news content and user behaviors into the news graph. Specifically, we construct three new types of collaborative relations for entities, *i.e.,* co-occurring in the same news, clicked by the same user and clicked by the same user in the same browsing session.
- Extensive experiments are conducted on a real news dataset. The results demonstrate the effectiveness of our news graph for multiple news recommendation tasks, including item recommendations, article category classification, article popularity prediction and local news detection.

2 NEWS GRAPH CONSTRUCTION

In this section, we introduce how to construct the NG in detail, including the construction of the news-relevant KG, collaborative relations and topic entities. We present an illustrative overview of NG in Fig. 1.

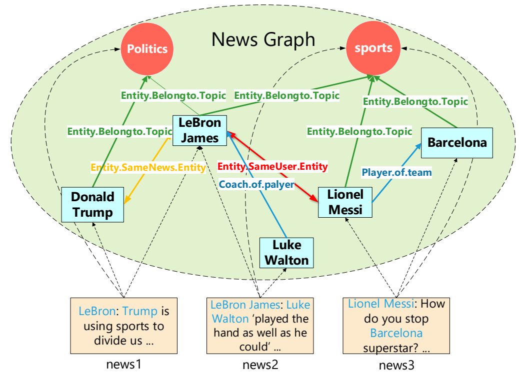

Figure 1: An overview of news graph construction. The blue arrows represent the relations in KG, *e.g., Luke Walton is the coach of LeBorn James (Coach.of.Player).* The red circles represent the news topic entities and green arrows are the topic relations between entities, *e.g., Donald Trump in news1 belongs to the topic Politics (Entitiy.Belongto.Topic).* The red arrows are the collaborative relations between entities that browsed or clicked by the same user,*e.g., LeBron James in news2 and Lionel Messi in news3 are clicked by the same user (Entity.SameUser.Entity).* The yellow arrows are the relations between entities in the same news, *e.g., Both LeBorn James and Donald Trump are reported in the news1 (Entity.SameNews.Entity).*

2.1 News-Relevant Knowledge Graph

We use a news corpus from MSN News[2] ranging from Nov. 2018 to Apr. 2019, which contains 621,268 news articles and 594,529 distinctive news entities. To incorporate the extra knowledge information, we adopt Microsoft Satori[3], which is a large scale commercial knowledge graph. For efficiency consideration, we search the one hop neighbors of all occurred entities in our news corpus in Microsoft Satori KG and extract all triples in which the confidence of relations linked among entities are greater than 0.8. The basic statistics of the extracted knowledge graph in our news corpus are shown in Table 1.

However, we observe that many relations in KG, *e.g.,* the *Birthday* of Donald Trump, may not be very relevant to news recommendation tasks, but lead to the increase of millions of irrelevant triples in NG (comparing the number of Triples in *KG* and *News-Relevant KG* in Table 1). Including enormous irrelevant relations not only makes the knowledge graph less effective to provide news-related information, but also makes some explainable recommender models such as [28] harder to search good knowledge paths for reasoning. we propose a News Relation Selecting (NRS) algorithm to filter out the news irrelevant relations. The details of NRS algorithm are shown in Algorithm 1 on the facing page. The basic idea of this

[3]https://searchengineland.com/library/bing/bing-satori

KG		
# Entities (1-hop)	# Relations	# Triples
3,392,942	2,681	46,048,763
News-Relevant KG		
# Relevant Entities	# Relevant Relations	# Relevant Triples
3,312,924	1000	43,119,590
Collaborative Relations in NG		
# Same News Triples	# Same User Triples	# Same Session Triples
2,111,918	17,465,043	7,255,555
Topic Entities in NG		
Topic Types	# Topic Entities	# Topic Triples
Category	704	376,624
LDA	1000	1,399,144

Table 1: The statistics of the news graph.

algorithm is that we search at most 2-hop neighbors in the above-mentioned KG for entities which appear in news articles (we called it *news entities* henceforth). During the search, if it reaches another news entity, we increase the weight for the relation that links these two news entities. At last, top relations with largest weight are considered as news related relations. The values of weight parameters are tuned manually according to observations and comparisons of the outcomes, and they are set to $w_1 = 1$ and $w_2 = 0.1$ finally.

Algorithm 1: Selection of News-Relevant Relations

Input: The knowledge graph before relation reduction: K_b; The news entity set: E_n; 1-hop relation weight: w_1; 2-hop relation weight: w_2; The number of relations to be reserved: n
Output: The knowledge graph after relation reduction: K_a

Relation Weight Set: $W_r = \varnothing$;
for t *in the original relation set* R **do**
    $W_r(t) = 0$ % Init all relation weight to 0
for e_i *in* E_n **do**
    Get e_i's 1-hop neighbor (relation:entity) set: N_i^1 ;
    for $(r_j : e_j)$ *in* N_i^1 **do**
        if e_j *in* E_n **then**
            $W_r(r_j) = W_r(r_j) + w_1$
        Get e_i's 2-hop neighbor (relation:entity) set: N_{ij}^2 via e_j;
        for $(r_k : e_k)$ *in* N_{ij}^2 **do**
            if e_k *in* E_n **then**
                $W_r(r_j) = W_r(r_j) + w_2$;
                $W_r(r_k) = W_r(r_k) + w_2$;
Sort W_r descending ;
$W_{rs} = W_r[1:n]$ %select top n relations ;
$K_a = \varnothing$;
for *triple in* K_b **do**
    if *triple.relation in* W_{rs} **then**
        K_a.add(triple) ;
Return K_a;

2.2 Enhanced with Collaborative Relations

Previous studies [25, 26] had proved that the rich information in KG can be utilized to alleviate the data sparsity and explainable problems, however, they are unaware of the collaborative relations of entities which are conveyed in the news content and clicking behaviors of users. Such collaborative relations are highly related to news recommendations and can be utilized to enhance our NG. We consider the three types of collaborative relations, *i.e.,* entities in the same news, entities clicked by the same user and entities appear in the same browsing session. The statistics of such relations are shown in *Collaborative Relations in NG* part in Table 1. For all the relations, we set the establish threshold to be ten times (*i.e.,* a relation between two entities is constructed when it appears over ten times).

Entities in the same news. Entities frequently co-occurring in the same news usually indicates that they are somehow related in news domain. *e.g.,* Lebron James and Donald Trump often occur in the same news due to their different political opinions (see in *news 1* in Figure1). This co-occurrence relation may reveal some hidden relationships of entities in news. We therefore add the relation, *i.e., Entity. SameNews. Entity*, between two entities.

Entities clicked by the same user. The entities clicked by the same user may imply some interest associations among them. For instance, in China there are many people who are fans of both *MayDay*[4] and *Jay Chou*[5]. These two entities are not directly connected in a generic knowledge, however, under the news graph context, they should be connected because if a user click on articles related to either one of them, there is a high probability that he/she will click on articles related to the other. We add *Entity. SameUser. Entity* relation between the entities clicked by the same user. See the example of *LeBron James* and *Lionel Messi* entities in Figure 1.

Entities appear in the same browsing session. This relation reflects the temporal correlation among entities. Given that a user has clicked some entities, we can infer what are the potential news he will click next (or in a short time) if the news graph is aware of the temporal relationship between knowledge entities. For instance, if a user has clicked on an article about weather forecast just now, then we should not recommend more weather forecast-related articles to him in the same session. However, if the last article a user clicked on is about a basketball player Kobe Bryant, then it is reasonable to recommend one more piece of news articles related to Kobe. This relation is especially useful in item-to-item recommendations.

2.3 Enhanced with Topic Entities

News topics are important factors to attract the attention of users. Not every news article contains knowledge entities. Sometimes users click on the article simply because they like the topics. To fill in the gaps where articles do not contain knowledge entities or contain non-informative entities, we propose to leverage news topics to make a supplement of the information of entities. We consider two types of topics information of news articles, *i.e.,* the explicit and implicit topics of articles. As classified by editors, the

[4]https://en.wikipedia.org/wiki/Mayday_(Taiwanese_band)
[5]https://en.wikipedia.org/wiki/Jay_Chou

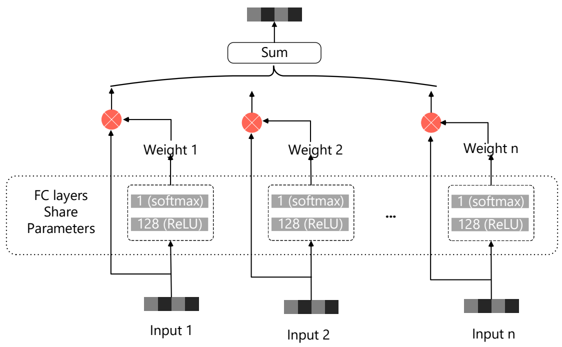

Figure 2: The architecture of attentive pooling component in all models.

category labels of articles are the best explicit topic information of articles. However, sometimes the simple category information may not be comprehensive enough to represent the topics of articles, especially when the articles do not have category labels. Hence we also utilize the Latent Dirichlet Allocation (LDA) model [15] to get the implicit topics of each article. We add the two types of news topic entities, *i.e.,* category and LDA entities, as the special entity nodes in NG. The linkage relations between an entity and its article topic entities are established only when the number of linkage time is over five times. Detailed statistics of topic entities are shown in *Topic Entities in NG* part in Table 1.

3 EXPERIMENTS

The goal of this paper is to propose a domain-specific knowledge graph for better news recommender systems. To demonstrate the effectiveness of the news graph, we design a series of simple experiments to compare the consumption of a general knowledge graph and our news graph. we conduct experiments on four typical news recommendation tasks, i.e. personalized article recommendations, news category classification, news popularity prediction and local news detection tasks. We use a real-world news reading dataset from MSN News[2] for experiments. We collect the user-item interaction logs from Jan.1, 2019 to Jan.28, 2019, which contain 24, 542 news articles, 665, 034 users, and a total number of 6,776,611 impressions.

3.1 Model Framework

For both general knowledge and our news graph, in order to consume knowledge entities, we adopt an attentive pooling component as depicted in Fig. 2 to merge all entities included in one news article into one embedding vector. The input of the attentive pooling component are entity embeddings learned from TransE [3], while the output is a merged knowledge-aware vector. Then the original document representation is enhanced by this merged knowledge-aware vector. In this paper, we mainly focus on verifying the effectiveness of NG, hence we adopt a simple but efficiency graph embedding method, i.e. TransE, to get the entity embedding vector in KG and NG for a fair comparison. Definitely we can explore other advanced methods to study the influence of graph embedding method in feature work.

Formally, suppose a news article n contains m entities $\{e_1, e_2, ..., e_m\}$. The original document vector (DV) of n is $\boldsymbol{v}_d$ (which can be generated by any models, such as DSSM [9] or BERT [5]). The proposed model framework is designed to generate another document representation $\boldsymbol{v}_n$ that contains the entity information, which can supplement the original document representation $\boldsymbol{v}_d$. Given a graph (KG or NG), we first adopt the widely used graph embedding method TransE [3] to obtain the entity embedding $\boldsymbol{v}_e \in \mathbb{R}^{D_e}$. To obtain a fixed-length vector representation $\boldsymbol{v}_+$ for entities, we aggregate the embedding of entities via the attentive pooling component. The normalized attention weight α_{e_j} of an entity e_j is defined as:

$$a_{e_j} = \boldsymbol{W}_2^{\mathrm{T}} \sigma(\boldsymbol{W}_1^{\mathrm{T}} \boldsymbol{v}_{e_j} + \boldsymbol{b}_1) + b_2, \quad (1)$$

$$\alpha_{e_j} = \frac{exp(a_{e_j})}{\sum_{k=1}^{m} exp(a_{e_k})}, \quad (2)$$

where a_{e_j} is the attention weight before normalization, which is computed by a two-layer attention network. $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are weight matrices of $\boldsymbol{v}_{e_j}$, b_1 and b_2 are the bias value in the two attention layers respectively. σ is the activation function, and we use ReLU in our model. Then the merged entity representation vector $\boldsymbol{v}_+$ is represented as:

$$\boldsymbol{v}_+ = \sum_{k=1}^{m} \alpha_{e_k} \cdot \boldsymbol{v}_{e_k}. \quad (3)$$

The new document vector $\boldsymbol{v}_n$ is computed by applying a non-linear transformation of the concatenated vector of $\boldsymbol{v}_d$ and $\boldsymbol{v}_+$:

$$\boldsymbol{v}_n = \sigma\big(\boldsymbol{W}_3^{\mathrm{T}}(\boldsymbol{v}_d \oplus \boldsymbol{v}_+) + \boldsymbol{b}_3\big) \quad (4)$$

where $\oplus$ denotes the concatenation of two vectors. In the following sections, a series experiments are designed to verify the superiority of the NG over the KG in learning useful $\boldsymbol{v}_n$. The overall architecture used for different tasks are illustrated in Fig. 3, more details will be introduced in the next section.

3.2 Recommendation Tasks

To verify the usefulness of NG for news recommendations, we conduct experiments on four news-related tasks:

- Personalized item recommendation task: given a user and a candidate article, it predicts the probability that the user will click on the article.
- Category classification task: to predict the category label[6] that a news article belongs to. We have 15 top-level categories in the news corpus, including US News, Entertainment, Sports, Lifestyle, Money, Celebrities, Royals News, World News, Travel, Autos, Politics, Health, Video, Weather and Food&Drink.
- Popularity prediction task: we split the articles into 4 balanced groups according to its click-through ratio (which indicates its popularity level). The task is to predict the popularity level given a news article.
- Local news detection task: it predicts whether a news article reports an event that happens in a local context that would not be an interest of another locality.

[6] In this task, the topic nodes, *i.e.,* category entities are not enabled in NG construction.

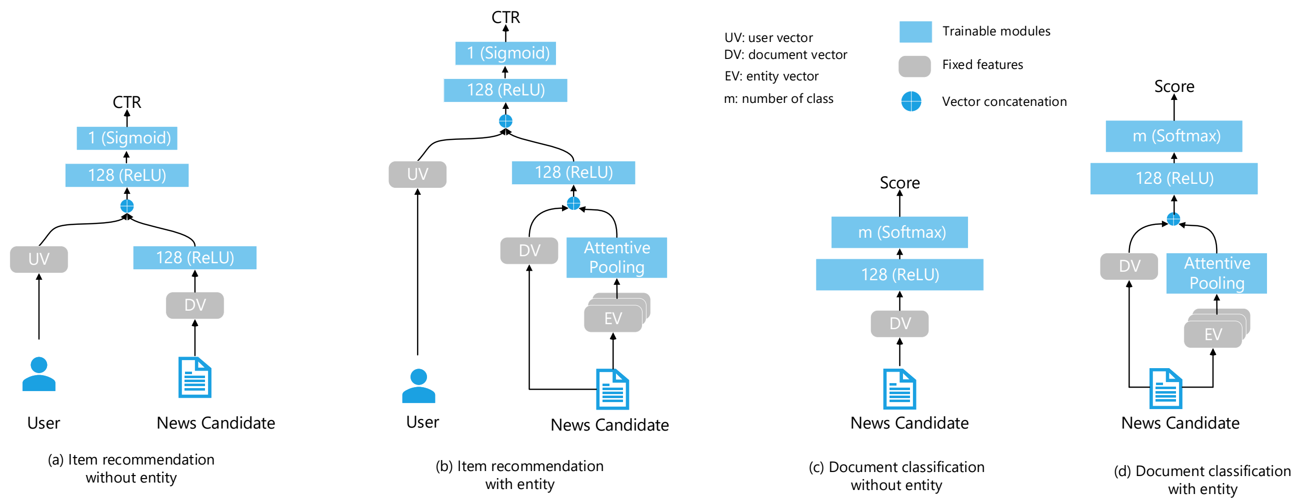

Figure 3: A summary of different model architectures for different tasks. (a) and (b) are for item recommendation task, while (c) and (d) are for news category classification, news popularity prediction and local news detection tasks.

For the item recommendation task, we compute the click through ratio (CTR) based on a concatenation of the user vector (UV) and the document vector (DV). UV is computed by a simple time-decayed averaging of DV of the user's clicked articles. The DV $\boldsymbol{v}_n$ is derived by the method described in Section 3.1. A two-layer feed-forward neural network is used to get the CTR prediction score. For optimization we use a ranking loss, *i.e.,* for each positive user-item pair, we randomly sample five negative items and to maximize the softmax likelihood of the positive pair. The running model architectures are shown in Fig.3(a,b). For the rest of tasks, we treat them as classification problems (binary classification for local news detection, multi-class classification for news category classification and news popularity prediction), and only take the document vector as input. The loss function is cross entropy [7]. The running model architectures are shown in Fig.3(c,d).

3.3 Evaluation Metrics

For news recommendation task, Area Under Curve (AUC) [4] and Normalized Discounted Cumulative Gain at rank k (NDCG@k) [10] is utilized to evaluate the model performance across seven days. For the multi-classification problem, including category classification and popularity prediction tasks, we adopt Accuracy (ACC) and F1-Score (micro) as the evaluation metrics. As for the binary local news detection task, we use three metrics AUC, F1-Score and ACC. For news recommendation task, we use the first two weeks' data to construct the user click history, the third week for training and the evenly split the data in last week for validation and test. For the other three tasks, we randomly split the news articles into: 8 : 1 : 1 for training, validation and testing respectively.

3.4 Parameter Settings

For each method, grid search is applied to find the optimal settings. We report the result of methods with its optimal hyperparameter settings. The dimensions of article embedding learned by topic model LDA [1] and learned by the DSSM model [9] are set to 90, and the embedding dimensions of entity and article are set to 90 for a fair comparison. The learning rate is set to 0.005 and batch size is 500. We already release the source code at https://github.com/danyang-liu/NewsGraphRec.

3.5 Results and Analysis

To verify the usefulness of our proposed NG, we compare the performance of the following models.

- KG: entity embeddings are learned from KG.
- NG: entity embeddings are learned on our constructed NG.
- DV: only using the original article vector $\boldsymbol{v}_d$ (which is learned from LDA model [15] and DSSM [9], we concatenate the vectors from these two models as the original document vector).
- DV+KG: we concatenate DV and KG vectors as the final document vector, *i.e.,* $\boldsymbol{v}_n$ based on KG.
- DV+NG: we concatenate DV and NG vectors as the final document vector, *i.e.,* $\boldsymbol{v}_n$ based on NG.

The results of the news recommendation task are shown in Table 2. The results of category classification, popularity prediction and local news detection tasks are presented in Table 3. With comparisons of all baseline methods, we have the following observations:

- In personalized article recommendation task, we demonstrate the performance of all methods on seven consecutive days. We can see that DV and DV+KG models have comparable performance. Overall DV+KG is better than DV model on both AUC and NDCG@10, but not consistently across the seven days. The model DV+NG performs best than all baseline models consistently on everyday. This indicates that the information learned in our proposed NG is really helpful for the prediction of users' news preference. Meanwhile, the results verify our assumption that a general knowledge

[7]https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html#cross-entropy

Methods		Day1	Day2	Day3	Day4	Day5	Day6	Day7	Overall
AUC	DV	0.6625	0.6734	0.6641	0.6777	0.6671	0.6579	0.6792	0.6699
	DV+KG	0.6604	0.6791	0.6678	0.6710	0.6627	0.6625	0.6857	0.6718
	DV+NG	**0.6761**	**0.6898**	**0.6794**	**0.6951**	**0.6859**	**0.6693**	**0.6929**	**0.6854**
NDCG @10	DV	0.2739	0.2655	0.2429	0.2766	0.2571	0.2352	0.2630	0.2636
	DV+KG	0.2802	0.2712	0.2488	0.2641	0.2476	0.2380	0.2696	0.2648
	DV+NG	**0.2892**	**0.2903**	**0.2641**	**0.2779**	**0.2654**	**0.2504**	**0.2745**	**0.2769**

Table 2: Performance comparisons of different methods on the personalized news recommendation task. Best performance is in boldface.

Tasks		Category Classification		Popularity Prediction		Local News Detection		
Methods		ACC	F1-Score	ACC	F1-Score	AUC	F1-Score	ACC
1	KG	0.3830	0.4700	0.2633	0.2806	0.6792	0.5653	0.6016
2	NG	0.4332	0.5315	0.2977	0.3627	0.8424	0.7530	0.8038
3	DV	0.6687	0.8180	0.3004	0.3787	0.8339	0.7942	0.8451
4	DV+KG	0.6808	0.8315	0.3063	0.3713	0.8301	0.7516	0.8332
5	DV+NG	**0.6997**	**0.8425**	**0.3185**	**0.3880**	**0.8483**	**0.7992**	**0.8503**

Table 3: Performance comparisons of different methods on three news-related tasks. Best performance is in boldface.

graph may contain uninformative relations or miss some domain specific relations/entities, thus is sub-optimal for news recommendation.

- In the category classification, popularity prediction, and local news detection tasks, KG is the weakest baseline, since it is not a news specific graph, and only contains the information of entities in news articles. NG performs better than KG substantially, especially on local news detection task. This indicates that our constructed NG are more powerful for news-related tasks. DV model, which contains the whole text information of news article, performs better than NG model in category classification and popularity prediction tasks, while loss advantages on local news detection task. This observation demonstrates except for the entity information, the contextual information of the whole text in news articles is also helpful for news-related tasks. The news entities in KG and NG can better reflect the local news information, while less relevant to the popularity of news articles. The combined model DV+KG performs better than DV in most cases, except for F1-Score evaluation on popularity prediction. DV+NG model, it achieves the best performance on all the metrics on all tasks, further indicating our proposed NG is effective for a wide range of news-related tasks.

4 RELATED WORK

Traditional recommender systems mostly suffer from several inherent issues such as data sparsity and cold start problems. To address the above problems, researchers usually incorporate extra information to increase the capacity of data and make a supplement to the current model. Our work is highly related with knowledge-enhanced recommendation, which had been intensively studied in recent years in alleviating the data sparsity and providing the explainability virtue for recommender systems.

The extra information in knowledge base, such as Freebase [2], Google Knowledge Graph [21] and Bing Satori [18], Knowledge Valut [6], YAGO [8], Probase [29], had been proved to be critical for many real-world tasks, such as question answering, document representation [20] and graph-based recommendation [19]. These knowledge graph are constructed from the large volume of noisy text data. For example, both Google Knowledge Graph and Bing Satori developed entity databases for hundreds of millions of entities [23]. They extracted entities from real world objects and concepts including people, places, books, movies, events and so on. An entity may have some properties and relationships to other entities. In addition, external knowledge graph contains much more fruitful facts and connections about items [2]. For example, CKE [31] proposes a general framework to jointly learn from the auxiliary knowledge graph, textual and visual information. As for news recommendation, DKN [26] is proposed to incorporate knowledge embedding and text embedding from news content. RippleNet [25] proposes to simulate how users' preferences propagate over the knowledge graph and uses a memory neural network to capture users' high-order preferences based on knowlege entities. It motivates several new projects which so far achieve state-of-the-art performance in knowledge graph-based recommendations [27].

Another related task is knowledge graph embedding, which has been extensively investigated in recent years. There are also a large body of relational approaches for modeling the relational patterns on knowledge graphs [3, 13, 22, 30]. As for aggregating the node representation in graph based models, previous studies learn the low-dimensional representations of graph vertices with preserving graph topology structure, node content, and other information. For example, GCN [11] utilizes a localized graph convolutions for a classification task. GAT [24] uses self-attention network for information propagation, which utilizes a multi-head attention mechanism to increase model capacity. GCN and GAT are popular architectures of the general graph networks.

The major difference between prior work and ours is that they directly use the generic knowledge graph, while in our work, we design a domain specific knowledge graph, called news graph. We utilize collaborative relations of entities (e.g.,entities frequently appear in same news articles or clicked by same users), and the topic context of news article to construct a more powerful graph for news recommendations.

5 CONCLUSION

In this paper, we demonstrate the necessity of using a domain specific graph for news recommendation, which yet has not been explored in the literature. We propose to construct a news graph by considering the collaborative relations and topic entities, as well as filtering out the irrelevant relations of news reading applications for efficiency consideration. Currently, we adopt a simple attention-based model to demonstrate the effectiveness of our proposed news graph. In the future, we will design more flexible and effective models to fuse the knowledge entities in a news article for precise document understanding. Meanwhile, we will explore how the news graph can benefit more tasks related to recommender systems, such as the model explainability [7] and item candidates retrieval [14].

ACKNOWLEDGMENTS

The authors would like to thank Microsoft News for providing technical support and data in the experiments, and Jiun-Hung Chen (Microsoft News) and Ying Qiao (Microsoft News) for their support and discussions.

REFERENCES

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

[2] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Sigmod Conference*.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*. 2787–2795.

[4] Andrew P Bradley. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition* 30, 7 (1997), 1145–1159.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. 4171–4186. https://aclweb.org/anthology/papers/N/N19/N19-1423/

[6] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. (2014).

[7] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. 2019. Explainable Recommendation through Attentive Multi-View Learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. 3622–3629.

[8] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence* 194 (2013), 28–61.

[9] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2333–2338.

[10] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[11] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[12] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2018. Towards better representation learning for personalized news recommendations: a multi-channel deep fusion approach. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 3805–3811.

[13] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion.. In *AAAI*.

[14] Zheng Liu, Yu Xing, Jianxun Lian, Defu Lian, Ziyao Li, and Xing Xie. 2019. A Novel User Representation Paradigm for Making Personalized Candidate Retrieval. *CoRR* abs/1907.06323 (2019). arXiv:1907.06323 http://arxiv.org/abs/1907.06323

[15] Zhiyuan Liu, Yuzhou Zhang, Edward Y. Chang, and Maosong Sun. 2011. PLDA+: Parallel Latent Dirichlet Allocation with Data Placement and Pipeline Processing. *ACM Transactions on Intelligent Systems and Technology, special issue on Large Scale Machine Learning* (2011). Software available at https://github.com/openbigdatagroup/plda.

[16] Zhongqi Lu, Zhicheng Dou, Jianxun Lian, Xing Xie, and Qiang Yang. 2015. Content-based collaborative filtering for news topic recommendation. In *Twenty-ninth AAAI conference on artificial intelligence*.

[17] Lenin Mookiah, William Eberle, and Maitrayi Mondal. 2018. Personalized news recommendation using graph-based approach. *Intelligent Data Analysis* 22, 4 (2018), 881–909.

[18] Richard Qian. 2013. Understand your world with bing. *Bing search blog, Mar* (2013).

[19] Yanru Qu, Ting Bai, Weinan Zhang, Jianyun Nie, and Jian Tang. 2019. An End-to-End Neighborhood-based Interaction Model for Knowledge-enhanced Recommendation. (2019).

[20] Michael Schuhmacher and Simone Paolo Ponzetto. 2014. Knowledge-based graph document modeling. (2014).

[21] Amit Singhal. 2012. Introducing the knowledge graph: things, not strings. *Official google blog* 5 (2012).

[22] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197* (2019).

[23] Ahmet Uyar and Farouk Musa Aliyu. 2015. Evaluating search features of Google Knowledge Graph and Bing Satori: entity types, list searches and query interfaces. *Online Information Review* 39, 2 (2015), 197–213.

[24] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[25] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 417–426.

[26] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1835–1844.

[27] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. ACM, New York, NY, USA, 950–958. https://doi.org/10.1145/3292500.3330989

[28] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5329–5336.

[29] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. (2012).

[30] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575* (2014).

[31] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *SIGKDD*.