IGNiteR: News Recommendation in Microblogging Applications

Yuting Feng
CNRS LISN
University Paris-Saclay
yuting.feng@universite-paris-saclay.fr

Bogdan Cautis
CNRS LISN and CNRS IPAL Singapore
University Paris-Saclay
bogdan.cautis@universite-paris-saclay.fr

2022 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-5099-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDM54844.2022.00111

***Abstract*—As social media, and particularly microblogging applications like Twitter or Weibo, gains popularity as platforms for news dissemination, personalized news recommendation in this context becomes a significant challenge. We propose a diffusion and influence-aware approach, Influence-Graph News Recommender (IGNiteR), which is a content-based deep recommendation model that jointly exploits all the data facets that may impact adoption decisions, namely semantics, diffusion-related features pertaining to local and global influence among users, temporal attractiveness, and timeliness, as well as dynamic user preferences. We perform extensive experiments on two real-world datasets, showing that IGNiteR outperforms the state-of-the-art deep-learning based news recommendation methods.**

***Index Terms*—News recommendation, deep learning, diffusion**

I. Introduction

News recommendation has been a topic of great interest in the field of recommender systems, and we have seen in recent years a plethora of ML-based techniques, such as [1]–[5]. Therein, additional challenges must be overcome, pertaining to the highly dynamic, ephemeral nature of news. However, in microblogging platforms like Twitter or Weibo, the release, dissemination, and adoption of news items follow a unique pattern, different from the ones of traditional news portals. A news item is firstly posted by a user (the cascade initiator), then it may attract the attention of his / her followers who may in turn repost (adopt) it. This news referral can continue and thus propagate to a large audience.

Word-of-mouth mechanisms allow information to propagate easily to a large audience. Moreover, they give credibility to the conveyed messages. Therefore, both the content of a news item and the users involved in its dissemination may determine how far that item may propagate and the extent of its adoption. Intuitively, when a news item reaches a user in the social platform, besides the explicit semantic information it exposes, time has endowed it with other, socially-related information that may sway that user's adoption decision.

Consequently, as microblogging gains popularity as a medium for news dissemination, *personalized news recommendations* in this scenario becomes an important challenge, one for which the information diffusion patterns and influence mechanisms therein must be well-understood and exploited for effective recommendations. Interestingly, social media is not only the second most important news source (behind TV), accounting for 40% of the consumption in recent statistics [6] but, in many ways, it has also reshaped the style of news and the users' patterns of news consumption.

In our view, the main limitation of existing ML-based news recommendation approaches is that they are generally based on the semantic content of news and on the user profiles, while the underlying recommendation scenario is ignored. We aim to address this limitation, in the *microblogging* context, for the news recommendation problem. There, a user may adopt a news item not only based on content / semantics, personal preferences, or timeliness of that news, but also based on the influence others may exert on her with regards to news adoption. Influence may be exerted locally (by friends / followees) or globally (indirectly, by highly influential users).

We propose a deep-learning based model, called IGNiteR, which requires in the training phase a joint history of news adoptions and news propagation traces (news cascades) from the microblogging application. IGNiteR seeks to exploit jointly all the data facets that may impact news adoption decisions, namely semantics, diffusion-related features pertaining to local and global influence among users, dynamic user preferences, as well as temporal attractiveness and timeliness.

Our main contributions are the following:

- We describe how to leverage diffusion cascades to build a behavior-driven user graph and node embeddings for the follow-up recommendation, revealing correlations among users and pinning down an estimate on the probability of information diffusion between them.
- We propose to incorporate the influence-level information, based on the participation of users in the news dissemination process, along with semantics, attractiveness, and timeliness for comprehensive news representation.
- We design IGNiteR, in which we attentively fuse the informativeness from different data facets through a news encoder, aggregating the news history with an attention-based sequential model for user profiling.

Our experiments with real-world datasets (including a publicly available one) show that IGNiteR outperforms the state-of-the-art deep-learning based news recommendation methods.

II. Related Work

We focus our related works discussion on the areas of social-aware or news recommendation techniques.

978-1-6654-5099-7/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDM54844.2022.00111

A. News Recommendation

Benefiting from recent NLP advances, many state-of-the-art works use pre-trained word embedding representations to address the problem of highly condensed semantic information. In [7], the authors explore the use of *pre-trained language models* (PLM) to mine the deep semantic information of news. Additional information such as *news categories* [5] or *dwell time* [8] have also been taken into consideration to represent news. External links to *knowledge-level information* [1] have broadened the scope of signals for predicting news adoption.

Based on the news representation, users are profiled through the aggregation of (some of) their adopted news, by various methods. For example, in [9], the authors exploit diverse recurrent networks to model the sequential evolution of user preferences, while [2] uses a Long Short-Term Memory (LSTM) network to encode clicked news. A GRU network is used in [10], which proposes to learn and use jointly long-term and short-term user representations. In [11], the authors propose a fine-grained interest matching method, where each news items in a user's history is endowed with multi-level representations via stacked dilated convolutions. Some recent studies adopt a *graph perspective*, building users–news graphs [4] or users–news–topics graphs [12] for recommendation.

The aforementioned models generally ignore the recommendation scenario, being generic by design. For news adoptions and recommendation in social platforms, relevant connections between users can be built not only based on commonly clicked news, but also based on *social connectivity* and, importantly, on (i) the implicit influence exerted among users and (ii) the observable diffusion patterns it leads to. To enhance the recommendation process, we place at the core of our model social graph embeddings for users, which can capture local and global influence inferred from information cascades.

B. Social-Aware Recommendation

The main underlying idea of recommender systems in social media is to capitalize on various social connectivity concepts, such as homophily and influence, and to extract correlations among users. Early recommender system models of this kind, not necessarily pertaining to news, leverage the social links as indicators of similarity in collaborative filtering approaches [13], [14]. Recent deep neural network (DNN) based research focuses on the latent representation of connectivity among users. For example, [15], [16] model social similarity by aggregating a user's social neighbours, leading to a social-space latent factor for users. In [17], the authors build a GNN to recursively update users and items embeddings, by exploring the social network up to a pre-defined depth. The recent work of [18] describes a social-aware recommendation model leveraging the social connections among users to build a user-user graph, complementing the user-item and item-item graphs constructed from the click history.

Beyond social similarity, understanding how information items may be diffused and adopted in sequence by socially connected users (i.e., influence) is of course paramount for effective recommendations in a social media context. Nevertheless, the reality of social media is much more intertwined, and the dissemination paths of news may not strictly follow the known social network topology (e.g., followership in Twitter). Indeed, users may be exposed to information published by others without direct connections. In our work, we adopt a deep behavior-driven network to model the social correlations among users by classifying them into influencers (news post initiators) and influencees (reposters), according to their participation in observed diffusion cascades of microblogging posts about news. We build user node embeddings from a given history of diffusion cascades, as the user representation for the subsequent news recommendation task.

III. Problem Formulation

Generally, in news recommendation, a given user i has an adoption history consisting of a set or a sequence of news items $x_1, x_2, ..., x_{s_i}$, and the recommendation task is to predict the probability that i will adopt (e.g., click on) some unseen candidate news x. In the microblogging context, the notion of click is replaced by the one of posting / tweeting. In such a practical context, a piece of news is first posted by the diffusion (cascade) initiator, and then adopted (posted) by other users involved in that diffusion process. When a cascade reaches a user, he or she will be notified that friends / followees adopted that news item, increasing awareness about it and thus influencing the adoption decision.

In our study, the overall input for our news recommendation framework consists of (i) a follower graph $G = (V, E)$, where V are the nodes (users) and E are the followship edges, possibly enriched by various node or edge features, and (ii) a cascade history $\mathcal{H}$, i.e., a set of diffusion cascades, which gives the timed news adoptions, possibly enriched by various news features. A diffusion cascade for item x is a time-ordered sequence of adoptions

$$C_x = [(v_0, t_0), (v_1, t_1), \ldots (v_{m^x}, t_{m^x})] \tag{1}$$

initiated by influencer v_0 at time t_0, with all v_j, $j \geq 1$, being the reposters and m^x denoting the number of reposters of x. From the cascades $\mathcal{H}$, we also distinguish the overall subsets of users $V_{influencer} \subseteq V$ and $V_{reposter} \subseteq V$.

When a target user i is exposed to a news item x at time t, several data facets of x may contribute to i's adoption decision: (i) the news content, (ii) the users already involved in the diffusion process of x up to moment t, (iii) the attractiveness of x, and (iv) the timeliness (lifespan) of x. More formally, a news item x at time t has as raw (initial) representation ϕ_t^x, which will be short notation for the tuple (s^x, V_t^x, A_t^x), where s^x is the semantic description of x, the sequence $V_t^x = [v_0, v_1, \ldots, v_{m_t^x}]$ represents the users who participated in the diffusion process of x up to moment t[1], with v_0 the initial user who published the news (initiator of the propagation chain), and A_t represents attractiveness (captured in our training data layout by histograms on the number of adoptions over time).

[1]A projection of C_x on the user dimension, up to moment t.

Therefore, the recommendation task becomes one of predicting whether a target user i will adopt at time t the candidate news x, described initially by ϕ_t^x, based on i's adoption history $\{\phi_{t_1}^{x_1}, \phi_{t_2}^{x_2}, \ldots, \phi_{t_{s_i}}^{x_{s_i}}\}$, where each $\phi_{t_j}^{x_j}$ represents the state of a news x_j when it was adopted by i at time t_j, $t_j < t$, with s_i being the length of i's relevant history.

IV. Our Approach

A. IGNiteR Influence-Aware User Graph Neural Network

Different from recommendation works where the social-connectivity induced features are mostly considered to build GNNs for node representations, we draw here inspiration from the recent work of [19], on spread modeling and maximization by node representations learned from cascades. We train a neural network in order to obtain embeddings for influencers (cascade initiators) – and for reposters of news, which will then be used for the subsequent recommendation task. (For further details on this dimension of our approach we refer the reader to the extended version of the paper [20].)

B. Personalized Cascade Views

Recall the initial description $\phi_{t_0}^x$ of a news item x at publication time t_0 is a tuple $(s^x, V_{t_0}^x, A_{t_0}^x)$, where $V_{t_0}^x = [v_0]$. As it spreads, by what can be seen as a snowball effect, the news item carries richer and richer information as more users get involved, such that its raw representation ϕ_t^x at time t becomes (s^x, V_t^x, A_t^x), with $V_t^x = [v_0, v_1, \ldots]$ now a potentially long time-ordered sequence of nodes. To select the most relevant users for the perspective of target user i, as well as to limit the computation cost, we keep the most influential nodes and the closest neighbors of i, either by connectivity in G or by similarity score from the reposter embedding matrix. (The detailed sampling procedure for i's perspective on an incoming cascade can be found in [20]). With the resulting refined node sub-sequence $V_t^i = \left[v_0, v_1^{'}, \ldots, v_m^{'}\right]$[2], $v_j^{'} \in V_t$, the raw description ϕ_t^x of news x will thus be given by the tuple (s^x, V_t^i, A_t^x). To simplify notation, whenever user i and news x are implicitly assumed in the following, the diffusion sequence will simply be denoted by $V_t = [v_0, v_1, \ldots, v_m]$ and the news information ϕ_t by (s, V_t, A_t).

C. Multi-Level Attention-based News Encoder

In order to capture not only semantics, but also the diffusion history and temporal information, we designed a *time-sensitive news encoder* to comprehensively encode the news items, starting from the raw description ϕ_t.

1) Semantic Information Extraction: For the semantic facet of news, we use a one-dimensional CNN [21] to extract the semantic information from text. We focus here on the news title (this can be easily extended to the abstract / article) seen as a sequence of words $s = \{w_1, w_2, \ldots, w_n\}$, with n denoting the title length. We use a pre-trained word embedding method, based on a large corpus, in order to get an overall word-embedding matrix $\boldsymbol{S} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n]^T \in \mathbb{R}^{n \times g_1}$, where g_1 is the word-embedding dimension.

The CNN uses multiple filters on the word-embedding matrix $\boldsymbol{S}$, each filter $\boldsymbol{h} \in \mathbb{R}^{g_1 \times l}$ being applied on the sub-matrix $\boldsymbol{S}_{i+l-1}$, with varying window size l, to get a new feature

$$c_i^h = f(\boldsymbol{h} \cdot \boldsymbol{S}_{i:i+l-1} + \boldsymbol{b}) \tag{2}$$

with $i = 1, 2, \ldots, n-l+1$ and the bias $\boldsymbol{b} \in \mathbb{R}_1^g$. As the filter goes in the direction of sentence length, a feature map $[c_1^h, c_2^h, \cdots, c_{n-l+1}^h]$ is obtained for each filter; then, by max-pooling, we can choose the most representative feature

$$c_{max}^h = \max\{c_1^h, c_2^h, \cdots, c_{n-l+1}^h\}. \tag{3}$$

The final encoding of news semantics is given by the concatenation of the max-pooling results of γ filters, denoted as

$$\boldsymbol{e^s} = [c_{max}^{h_1}, c_{max}^{h_2}, \cdots, c_{max}^{h_\gamma}]. \tag{4}$$

2) Diffusion Cascade Aggregation: Given the node sequence $V_t = [v_0, v_1, \ldots, v_j, \ldots, v_m]$ obtained as in Sec. IV-B, the node embedding sequence (detailed in Sec. IV-A) can be written as $\boldsymbol{V} = [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots \boldsymbol{v}_m]^T \in \mathbb{R}^{m \times g_2}$, where g_2 denotes the node embedding dimension. To capture temporal correlations and dependencies among users, we use an LSTM model as the encoder of V_t, since V_t is a sequence of time-ordered users. Considering the node sequence as an input of m time steps, each LSTM cell can be computed as follows:

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f\,[\boldsymbol{h}_{t-1}, \boldsymbol{v}_t] + \boldsymbol{b}_f) \tag{5}$$
$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i\,[\boldsymbol{h}_{t-1}, \boldsymbol{v}_t] + \boldsymbol{b}_i) \tag{6}$$
$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o\,[\boldsymbol{h}_{t-1}, \boldsymbol{v}_t] + \boldsymbol{b}_o) \tag{7}$$
$$\widetilde{\boldsymbol{C}}_t = \tanh\left(\boldsymbol{W}_C\,[\boldsymbol{h}_{t-1}, \boldsymbol{v}_t] + \boldsymbol{b}_C\right) \tag{8}$$
$$\boldsymbol{C}_t = \boldsymbol{f}_t \odot \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \odot \widetilde{\boldsymbol{C}}_t \tag{9}$$
$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh \boldsymbol{C}_t \tag{10}$$

where $\boldsymbol{h} \in \mathbb{R}^u$ is the hidden state, $\boldsymbol{W}_f$, $\boldsymbol{W}_i$, $\boldsymbol{W}_o$, $\boldsymbol{W}_C \in \mathbb{R}^{(u+g_2) \times u}$ are the weighted matrices and $\boldsymbol{b}_f$, $\boldsymbol{b}_i$, $\boldsymbol{b}_o$, $\boldsymbol{b}_C \in \mathbb{R}^u$ are the biases of the LSTM, trained to parameterize the forget, input, output gates, and block input respectively. u denotes the number of units in an LSTM cell.

In addition, as the influence that each node in V_t may exert on the target user i may vary considerably, instead of getting the final step, we retain the hidden state of each time-step into a sequence $[\boldsymbol{h}_0, \boldsymbol{h}_1, \cdots, \boldsymbol{h}_m]$, and apply an attention mechanism to aggregate the output of LSTM cells:

$$h_i = \boldsymbol{q}_h{}^T \tanh(\boldsymbol{W}_h \boldsymbol{h}_i + \boldsymbol{b}_h) \tag{11}$$

$$\alpha_i^h = \mathrm{softmax}(h_i) = \frac{\exp(h_i)}{\sum_{j=0}^m \exp(h_j)} \tag{12}$$

where $\boldsymbol{W}_h \in \mathbb{R}^{u \times u}$, $\boldsymbol{b}_h \in \mathbb{R}^u$ are the projection parameters, and $\boldsymbol{q}_h \in \mathbb{R}^u$ is the attention query vector. With the attention weights, the node sequence is expressed as the weighted representation of the hidden state from each time step

$$\boldsymbol{e}^v = \sum_{j=0}^m \alpha_i^h \boldsymbol{h}_j. \tag{13}$$

[2]Here, the superscript i replaces the x one, to denote user i's perspective on the cascade chain of news item x.

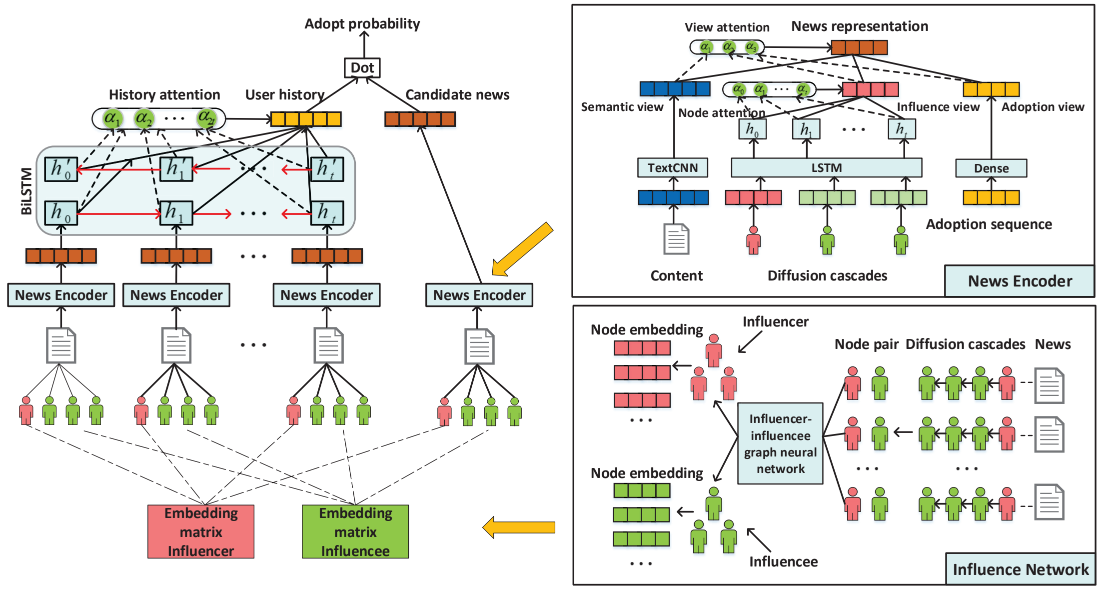

Fig. 1. The framework of IGNiteR.

3) News Adoption Sequence and Lifespan: The distribution of the number of adoptions over time can be an indicator of popularity and evolution trend – what we call *attractiveness*, while the timeliness of news can be captured by the lifespan since the initial posting. Moreover, as the adoption chart traces the adoption evolution pattern of a news item, differences in the target user's reaction to such a news item at a specific moment in the charted evolution may be indicative of different news types / categories and preferences thereof [22].

We use the sequence $A_t = [(a_0, t_0), (a_1, t_1), \ldots, (a_d, t_d)]$ of adoptions, where a_i records the number of adoptions (retweets) during a time step, and t_i is the corresponding timestamp. Considering that the number of adoptions can vary significantly and may be very large in certain cases, and that news popularity decays with time, we calibrate the raw counts a_i as $a_i^{'} = f(a_i)(t_d - t_0)^{-1}$, where f is the scaling function, t_d is the current moment, and t_0 is the publication time of the item. The adoption vector becomes $e^a = \left[a_1^{'}, a_2^{'}, \ldots, a_d^{'}\right]$.

4) Multi-Views Attentive Fusion: In the end, we propose a macro-view attention network to consolidate the three components from different sources. Firstly, we align the dimensions of the three vectors $[e^s, e^v, e^a]$ to the same vector space $\mathbb{R}^g$, through linear / non-linear transformations for e^s and e^v, while padding is used for the adoption sequence e^a. Then, we build an attention network to allocate weights to the semantics information, influence representation, and adoption evolution pattern respectively, noted as α_s, α_v, and α_a. For illustration, the semantics weight is computed with:

$$a_s = \boldsymbol{q_s}^T \tanh(\boldsymbol{W}_s \boldsymbol{e}_s + \boldsymbol{b}_s) \tag{14}$$

$$\alpha_s = \frac{\exp(a_s)}{\exp(a_s) + \exp(a_v) + \exp(a_a)}, \tag{15}$$

where $\boldsymbol{W}_s \in \mathbb{R}^{g\times g}$ and $\boldsymbol{b}_h \in \mathbb{R}^g$ are the projection parameters, while $\boldsymbol{q}_s \in \mathbb{R}^g$ is the attention query vector. The attention weights for other views can be obtained in a similar way, and the final weighted encoding of a news item is as follows:

$$\boldsymbol{e}^n = \alpha_s \boldsymbol{e}^s + \alpha_v \boldsymbol{e}^v + \alpha_a \boldsymbol{e}^a \tag{16}$$

D. Time-Sensitive User Encoder

With the news encoder, we obtain the adoption history as a time-ordered sequence of news items $\{e^1, e^2, ..., e^s\}$, illustrating the evolution of user's preferences over time. Unlike a diffusion cascade, where the propagation is uni-directional, a user may still be interested in a topic that appears early in the history. To represent this, we adopt a bidirectional LSTM (BiLSTM) to use the "past" for "future" in the forward phase and vice-versa in the backward phase of the training.

The BiLSTM mechanism is similar to the LSTM, except that the hidden state output is doubled, so information from both ends is preserved. We thus obtain the hidden state sequence as $[\boldsymbol{h}_0, \boldsymbol{h}_1, \cdots, \boldsymbol{h}_{2s}]$.

To deal with the potentially diverse range of interests of users, an attention network is added upon the hidden state of the BiLSTM, to get the final weighted encoding

$$h_i^B = \boldsymbol{q}_B{}^T \tanh(\boldsymbol{W}_B \boldsymbol{h}_i^B + \boldsymbol{b}_B) \tag{17}$$

$$\alpha_i^{h^B} = \mathrm{softmax}(h_i^B) = \frac{\exp(h_i)}{\sum_{j=1}^{2s} \exp(h_j^B)}, \tag{18}$$

where $\boldsymbol{W}_B \in \mathbb{R}^{2s\times 2s}$ and $\boldsymbol{b}_B \in \mathbb{R}^{2s}$ are the projection parameters, while $\boldsymbol{q}_B \in \mathbb{R}^{2s}$ is the attention query vector. The final user encoding becomes the following:

$$\boldsymbol{e}^u = \sum_{j=1}^{2s} \alpha_i^h \boldsymbol{h}_j^B. \tag{19}$$

E. Model Training

In the training phase, we generate $1 + \lambda$ news items as an impression, among which one item is sampled from the user's history as a positive sample, and the remaining λ items are negative samples. Then, the recommendation task is reformulated as a $1 + \lambda$ multi-class classification. With the softmax function to normalize the adoption probability for each "class", the final adoption probability on the positive sample is expressed as follows:

$$p_i = \frac{\exp(\hat{y}^+)}{\exp(\hat{y}^+) + \sum_{i=1}^{\lambda} \exp(\hat{y}_i^-)}, \tag{20}$$

where $\hat{y}$ is the inner product between the sample news representation and the user representation, while $+$ denotes the positive sample and $-$ denotes the negative ones in the session. The loss function in the classification-like training becomes the minimization of the log-likelihood on all the positive samples:

$$\mathcal{L} = -\sum_{i=1}^{s} \log(p_i), \tag{21}$$

s being the number of positive samples, i.e., the history length.

V. Experiments

We used data from the main microblogging platforms, Twitter and Weibo, where users post and interact with messages that we generically call "tweets". We collected the Twitter data through its API, while the Weibo dataset comes from [23]. As news appear implicitly in tweets as links, pointing to the original publisher page, we identify these links in tweets, and then we crawl the news articles. Following some preprocessing and filtering steps (see [20] for more details), the statistics for the Twitter and Weibo datasets are respectively (i) number of users 248,195 and 692,833, (ii) number of retweets records 4,999,535 and 31,211,347, (iii) number of original tweets 4,566,942 and 232,978, (iv) number of news 441,632 and 13,513, (v) average number of words per title 6.94 and 7.26, (vi) median length of diffusion chain 2.74 and 23, (vii) maximal title length of diffusion chain 124 and 31009.

The IGNiteR code and the Weibo data can be found at: https://github.com/YutingFENG-5423/IGNiteR.

A. Comparison Models

We compare with 3 groups of state-of-the-art methods, thoroughly fine-tuned in order to obtain their best performance. For the group of *Generic Recommendation Models*, we use the classical factorization model **LibFM** [24], and deep-learning based CTR models **DeepWide** [25], **DeepFM** [26], **DCN** [27]. For the group of *Deep Neural News Recommendation Models*, we use **DKN** [1] and **DAN** [2] to explore the knowledge aspect, **NAML-BERT** [7] with pre-trained language model, **GERL** [4] for the graph representation of users and news, **LSTUR** [10] for long-term and short term user profiling, **NAML** [28] on the semantics and news categories, and **FIM** [11] as a fine-grained interest matching method. We selected **DICER** [18] as a method of the *Social-aware Recommendation Model* group, to explore the social dimensions of news adoptions. An ablation study was also carried out by removing from the news raw representation either the diffusion sequence V_t (**IGNiteR-V$_\mathbf{t}$**) or the adoption sequence A_t (**IGNiteR-A$_\mathbf{t}$**). See [20] for details on the implementation the baselines.

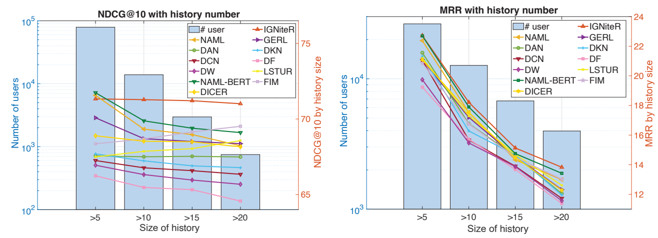

Fig. 2. NDCG@10 in Weibo (left) / MRR in Twitter (right) by history size.

B. Experimental Setting

We split the train / test data by the timeline, so that around 85% data is used for training and 15% for testing, while 10% from the train set is used for validation. For each impression in the training, the ratio positive-negative samples is 1 to 4. We set the count for representative nodes m to 30. The maximal number of adopted news items per user is set to 20, the maximum title length is set to 20, the adoption time unit is set to one hour, the maximum adoption length is set to 120, and the maximum number of negative samples in testing phase is set to 10. Considering the highly skewed adoption patterns for popular news, we use *two-times log* as the scaling function. The related hyperparameters can be found in [20].

C. Results Analysis

1) Comparison with baselines: We can note in Table I that IGNiteR outperforms the other methods on all metrics on microblogging news data, which validates our initial motivation.

We can see that, generally, LibFM performs worse than other deep learning-based models. NAML-BERT and FIM outperform the other models in Weibo and Twitter respectively. With a pre-trained language model, NAML-BERT improves the performance of NAML on both datasets, showing the effectiveness of a pre-trained language model in news recommendation. FIM proves the importance of fine-grained pairwise multi-level matching between candidate news and a user history (instead of a vector-wise user history view). We can also note that, although DICER is not specifically designed for news recommendation, it performs relatively well, compared to other general recommendation models; this further supports our initial motivation of enhancing news recommendation by the integration of social-aware information.

The results of the ablation study of IGNiteR show that both the users involved in the diffusion chain and the adoption pattern play an important role for our recommendation task, with the former contributing slightly more than the latter.

2) Variation with the history length: The size of the users' history in a given time window indicates how active and adoption-prone users may be, which can lead to variations in performance. To exploit this, we did separate experiments for user groups having different activeness levels. We divided the users into four groups according to their overall history, as users with more than 20, 15, 10, and 5 adopted items. The user count in each group is given in the histograms of Fig. 2, and we measure NDCG@10 for Weibo and MRR for Twitter.

Fig. 2 (left) shows the variation of performance with activeness. For the group of user with more than 5 adoptions,

TABLE I
COMPARISON AMONG IGNITER VARIANTS IN *Weibo* & *Twitter*

Models	Weibo				Twitter			
	AUC	MRR	NDCG@5	NDCG@10	AUC	MRR	NDCG@5	NDCG@10
LibFM	65.01 ± 1.34	13.47 ± 0.89	60.79 ± 1.10	61.34 ± 0.57	65.50 ± 1.05	10.25 ± 1.09	59.78 ± 0.60	61.45 ± 0.67
DeepFM	65.49 ± 1.13	15.12 ± 0.95	61.96 ± 1.28	64.57 ± 0.98	68.23 ± 0.82	11.38 ± 1.25	63.47 ± 0.98	65.76 ± 0.98
DeepWide	66.96 ± 0.78	15.79 ± 0.81	62.93 ± 1.21	65.68 ± 0.88	68.52 ± 1.05	11.57 ± 0.82	63.61 ± 0.62	65.88 ± 0.59
DCN	67.12 ± 0.96	17.78 ± 0.89	64.67 ± 0.34	66.34 ± 0.67	68.89 ± 0.91	11.73 ± 0.91	63.97 ± 0.76	66.02 ± 1.19
DKN	68.03 ± 1.32	18.01 ± 0.85	65.25 ± 1.57	66.98 ± 1.11	69.17 ± 1.15	11.96 ± 0.50	65.02 ± 0.27	67.08 ± 1.43
DAN	68.79 ± 0.79	18.27 ± 0.35	65.45 ± 0.77	67.49 ± 0.82	69.72 ± 0.68	12.05 ± 0.91	65.35 ± 0.86	67.18 ± 1.05
GERL	69.90 ± 1.02	18.42 ± 0.63	66.35 ± 0.96	68.37 ± 1.07	70.21 ± 0.38	12.32 ± 0.34	65.62 ± 0.59	67.74 ± 0.52
NAML	71.13 ± 1.54	19.06 ± 0.34	66.76 ± 2.08	68.23 ± 1.47	70.35 ± 0.69	12.99 ± 0.24	66.83 ± 1.22	69.25 ± 0.89
NAML-BERT	71.74 ± 1.04	19.44 ± 0.26	67.15 ± 0.75	69.01 ± 1.47	$\underline{71.05 \pm 0.33}$	$\underline{13.52 \pm 0.97}$	$\underline{67.20 \pm 1.45}$	$\underline{69.80 \pm 1.52}$
LSTUR	71.02 ± 0.87	19.00 ± 0.91	66.95 ± 0.38	68.57 ± 0.65	70.15 ± 0.57	12.61 ± 1.05	66.54 ± 1.77	68.57 ± 1.58
FIM	$\underline{72.69 \pm 0.94}$	$\underline{19.87 \pm 0.67}$	$\underline{67.56 \pm 2.03}$	$\underline{69.52 \pm 1.74}$	70.68 ± 0.89	13.02 ± 0.81	67.01 ± 0.96	69.65 ± 0.67
DICER	69.35 ± 0.99	17.92 ± 0.68	66.54 ± 1.20	68.16 ± 0.67	70.18 ± 0.74	12.28 ± 0.56	65.63 ± 1.22	67.75 ± 0.24
IGNiteR-A_t	73.34 ± 0.88	21.03 ± 0.34	68.77 ± 1.45	70.56 ± 0.57	71.93 ± 0.79	13.31 ± 0.17	69.09 ± 0.58	71.74 ± 0.87
IGNiteR-V_t	72.98 ± 1.37	20.23 ± 0.67	68.06 ± 1.13	70.07 ± 1.75	71.59 ± 1.02	13.07 ± 0.33	68.76 ± 1.02	70.93 ± 0.67
IGNiteR	$\mathbf{74.57 \pm 0.93}$	$\mathbf{21.57 \pm 0.25}$	$\mathbf{69.23 \pm 0.84}$	$\mathbf{71.11 \pm 0.79}$	$\mathbf{72.71 \pm 1.19}$	$\mathbf{13.83 \pm 0.58}$	$\mathbf{69.75 \pm 0.85}$	$\mathbf{72.03 \pm 0.25}$

NAML / NAML-BERT outperform slightly IGNiteR, while the performance of NAML and GERL decreases drastically for more active users. The overall curve of IGNiteR is slightly decreasing as well, but stabilises at a good level. The performance of FIM and LSTUR goes up as the activeness of users increases. It is interesting to note that DAN has a similar evolution and robustness, even though it generally performs worse than NAML and GERL. We can credit this to the LSTM / GRU component present in DAN, LSTUR, and IGNiteR, capturing dependencies in long sequences. For FIM, our interpretation is that its multi-level matching structure may be more suitable for picking the salient features when abundant data is available. Fig. 2 (right) shows that with increased history size, the increased informativeness makes the models progressively incapable of placing the positive samples in a conspicuous position. Hence the performance degrades for all models on the MRR metric; nevertheless, IGNiteR still outperforms the other models in all groups of users.

VI. CONCLUSION

We propose in this paper the IGNiteR content-based deep learning model for news recommendation, tailored for recommendation scenarios in social media. To incorporate awareness about news due to social influence, we represent users by embeddings obtained by methods leveraging the diffusion history (cascades), in such a way that news are endowed with diffusion-related information. A CNN method is applied to deal with the joint representation of news and an attention mechanism allows us to aggregate the users' diverse interests with respect to candidate news. Experiments were conducted on real-world datasets, showcasing the significant improvements of IGNiteR over SOTA recommendation models.

VII. ACKNOWLEDGEMENT

Yuting Feng is supported by the China Scholarship Council and by the CNRS FairIM@Scale project.

REFERENCES

[1] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: Deep knowledge-aware network for news recommendation," in *WWW*, 2018.

[2] Q. Zhu, X. Zhou, Z. Song, J. Tan, and L. Guo, "DAN: Deep attention neural network for news recommendation," in *AAAI*, 2019.

[3] L. Meng, C. Shi, S. Hao, and X. Su, "DCAN: Deep co-attention network by modeling user preference and news lifecycle for news recommendation," in *DASFAA*, 2021.

[4] S. Ge, C. Wu, F. Wu, T. Qi, and Y. Huang, "Graph enhanced representation learning for news recommendation," in *WWW*, 2020.

[5] C. Wu, F. Wu, M. An, J. Huang, Y. Huang, and X. Xie, "NPA: Neural news recommendation with personalized attention," in *KDD*, 2019.

[6] R. Institute. (2020) Executive Summary and Key Findings of the 2020 Report. [Online]. Available: https://www.digitalnewsreport.org/survey/2020/overview-key-findings-2020

[7] C. Wu, F. Wu, T. Qi, and Y. Huang, "Empowering news recommendation with pre-trained language models," in *SIGIR*, 2021.

[8] ——, "User modeling with click preference and reading satisfaction for news recommendation," in *IJCAI*, 2020.

[9] S. Okura, Y. Tagami, S. Ono, and A. Tajima, "Embedding-based news recommendation for millions of users," in *KDD*, 2017.

[10] M. An *et al.*, "Neural news recommendation with long-and short-term user representations," in *ACL*, 2019.

[11] H. Wang, F. Wu, Z. Liu, and X. Xie, "Fine-grained interest matching for neural news recommendation," in *ACL*, 2020.

[12] L. Hu, C. Li, C. Shi, C. Yang, and C. Shao, "Graph neural news recommendation with long-term and short-term interest modeling," *Information Processing & Management*, 2020.

[13] G. Guo, J. Zhang, and N. Yorke-Smith, "Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *AAAI*, 2015.

[14] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *WSDM*, 2011.

[15] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *WWW*, 2019.

[16] W. Xiao, H. Zhao, H. Pan, Y. Song, V. W. Zheng, and Q. Yang, "Beyond personalization: Social content recommendation for creator equality and consumer satisfaction," in *KDD*, 2019.

[17] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, and M. Wang, "Diffnet++: A neural influence and interest diffusion network for social recommendation," *IEEE TKDE*, 2020.

[18] B. Fu, W. Zhang, G. Hu, X. Dai, S. Huang, and J. Chen, "Dual side deep context-aware modulation for social recommendation," *arXiv:2103.08976*, 2021.

[19] G. Panagopoulos, F. Malliaros, and M. Vazirgiannis, "Multi-task learning for influence estimation and maximization," *IEEE TKDE*, 2020.

[20] Y. Feng and B. Cautis, "IGNiteR: News recommendation in microblogging applications (extended version)," *arXiv:2210.01942*.

[21] Y. Kim, "Convolutional neural networks for sentence classification," in *EMNLP*, 2014.

[22] C. Castillo, M. El-Haddad, J. Pfeffer, and M. Stempeck, "Characterizing the life cycle of online news stories using social media reactions," in *CSCW*, 2014.

[23] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li, "Social influence locality for modeling retweeting behaviors," in *IJCAI*, 2013.

[24] S. Rendle, "Factorization machines with LibFM," *ACM TIST*, 2012.

[25] H.-T. Cheng *et al.*, "Wide & deep learning for recommender systems," in *Workshop on deep learning for recommender systems*, 2016.

[26] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: a factorization-machine based neural network for CTR prediction," *arXiv:1703.04247*.

[27] R. Wang, R. Shivanna, D. Cheng, S. Jain, D. Lin, L. Hong, and E. Chi, "DCN-M: Improved deep & cross network for feature cross learning in web-scale learning to rank systems," *arXiv:2008.13535*.

[28] C. Wu, F. Wu, M. An, J. Huang, Y. Huang, and X. Xie, "Neural news recommendation with attentive multi-view learning," in *IJCAI*, 2019.