News Popularity Prediction with Local-Global Long-Short-Term Embedding

Shuai Fan[1], Chen Lin[1(✉)], Hui Li[1], and Quan Zou[2]

[1] School of Informatics, Xiamen University, Xiamen, China
chenlin@xmu.edu.cn
[2] Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China

Abstract. Predicting news popularity is an essential topic in the news industry. It is challenging because numerous factors influence public response to the news. This paper presents F^4, a neural model to predict news popularity by learning news embedding from global, local, long-term and short-term factors. F^4 integrates a sentence encoding module to represent the local context of each news story; a heterogeneous graph-based module to capture the short-term information propagation from current buzz words to each news story; a sequential module to extract long-term popularity features in entity sequence; and an attention module to learn global news-entity correlations. Extensive experiments on real-world Chinese and English news datasets demonstrated that F^4 outperforms state-of-the-art baselines in predicting and ranking news popularity.

1 Introduction

The development of the Internet has revolutionized the news industry. We have seen news agencies starting online news portal services and online news apps persistently boosting audiences. Predicting news popularity (i.e., estimating how many people will read a particular piece of news) becomes an important topic, and it builds the foundation for a broad spectrum of downstream tasks. For example, based on the predicted news popularity, online news portals can optimize the page layout and resource management; advertisers can tailor ads and save costs; news recommender systems can improve recommendation quality for news users, to name a few.

Recent years have witnessed numerous research in predicting the popularity of online news, which is measured by various user behaviors, including the number of likes [12,28], number of clicks [4], number of comments [21], number of instant messages [15], and so on. Conventional approaches are built upon hand-crafted features, including context features and news content features [4]. They apply shallow learning models, such as Support Vector Machines [4] and Logistic Regression [2]; topic models, such as named entity topic model [1]; and

W. Zhang et al. (Eds.): WISE 2021, LNCS 13081, pp. 79–93, 2021.
https://doi.org/10.1007/978-3-030-91560-5_6

statistical sequence models, such as the neural hawks process [13]. As features are costly to obtain and sometimes not accessible, modern distributed learning, which represents each news as a numerical vector called news embedding, has been applied in related problems such as news recommendation [26] and stock price movement prediction [11].

However, news popularity is not easy to predict since many factors (i.e., global, local, long-term, and short-term factors) influence how online users respond to a piece of news. We illustrate two example news in Fig. 1. We can see that, *news1* is more popular than *news2*, because (1) the content of *news1* (i.e., **local** factors) contains terms that are current buzz words (i.e., **short-term** factors) (2) *news1* is related, with different association strengths (i.e., **global** factors), to named entities that are consistently popular over time (i.e., **long-term** factors). In this paper, we propose F^4: a news embedding learning model from local, global, long-term, and short-term factors to predict news popularity. (1) For long-term factors, F^4 adopts a Gated Recurrent Unit (GRU) module on entity sequence to learn the sequential popularity embedding. (2) For the short-term factors, F^4 adopts a heterogeneous graph-based module to propagate information from current buzz words to the news. (3) For the local factors, F^4 adopts a sentence encoder to represent the local context of each news story. (4) For the global factors, F^4 adopts the attention mechanism to associate the popularity of each news story with different entities. (5) F^4 incorporates the above modules in a unified framework to make predictions by integrating all factors.

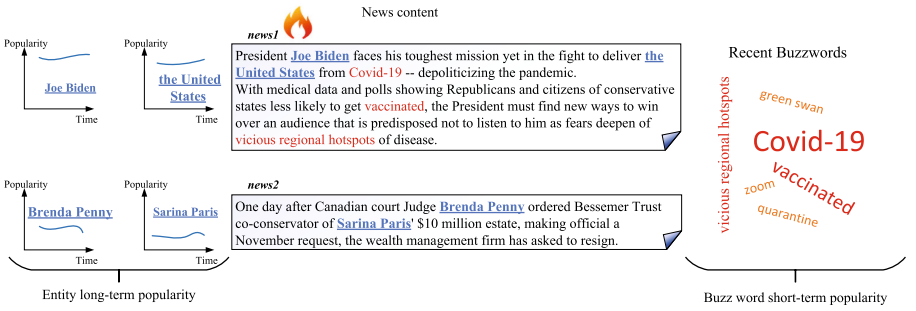

Fig. 1. Global, local, long-term and short-term factors affect news popularity. A popular news is related to long-term popular entities and the news content contains current buzz words.

In summary, our contributions are: (1) We propose to predict news popularity by integrating local, global, long-term, and short-term factors. (2) We develop neural models to learn local contextual representation, short-term popularity propagation, long-term sequential entity popularity, and global news-entity associations. (3) We demonstrate the effectiveness of our proposed method through extensive experiments on real-world Chinese and English datasets.

2 Related Work

With the development of online social networks, predicting the popularity of online news receives much attention and has been an active research area. Existing research generally falls into two categories. The first type forecasts news popularity based on textual contents. Clustering method [14], regression method [8,9], and classification method [22], have been proposed. Tatar et al. [22] used three models, including a linear model, a linear model on a logarithmic scale, and a constant scaling model to predict online news popularity. Stoddard et al. [20] proposed a simple Poisson regression model to estimate the quality of articles and found that news with higher popularity on Reddit and Hacker News is, to a large extent, articles with higher quality. The second type focuses on exploiting context features [4] and multi modal features [12].

The majority of previous research used the non-neural network models such as SVM classifiers [4], logistic regression [2], Tree Regression [6], etc. Recently, deep neural network which automatically extract vectorized numerical feature representations have shown promising results, including Convolutional Neural Network (CNN) [19], Recurrent Neural Network (RNN) [10], Long Short-Term Memory network(LSTM) [29], Graph Neural Network (GNN) [5], Autoencoder [18], and the Attention Mechanism [16]. We have seen applications of neural network models in similar tasks such as dwell time prediction [3] and news recommendation [27].

However, previous studies ignore the complex factors that influence the popularity of news. As we show in the following sections, purely relying on local content and missing the long-term, global entity information leads to low prediction accuracy. On the contrary, F^4 obtains high prediction accuracy by capturing different factors and their influence on news popularity.

3 Model

3.1 Preliminaries

Problem Definition. Suppose we have a training set $\mathcal{D}$, where each training instance is a tuple $< \mathbf{x}, y > \in \mathcal{D}$. The input for each news story is $\mathbf{x} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{N(\mathbf{x})}\}$, which contains $N(\mathbf{x})$ sentences. Each sentence is a sequence of words, i.e. $\mathbf{s}_i = \{\mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w}_{M(\mathbf{s}_i)}\}$, where $M(\mathbf{s}_i)$ is the length of $\mathbf{s}_i$. The output label y is the normalized public response to $\mathbf{x}$, i.e., $y \in (0, 1)$ (more details in Sect. 4). Our goal is to forecast $\hat{y}$ for any test instance $\hat{\mathbf{x}}$.

Model Overview. Figure 2 presents the framework of F^4, which mainly consists of four parts. (1) The **sentence encoding module** encodes the context within each sentence of a news story to represent the local factor of popularity prediction. (2) The **heterogeneous graph-based news encoding module** encodes the information propagation from current buzz words to each news story, based on a heterogeneous graph of words and sentences, to represent the

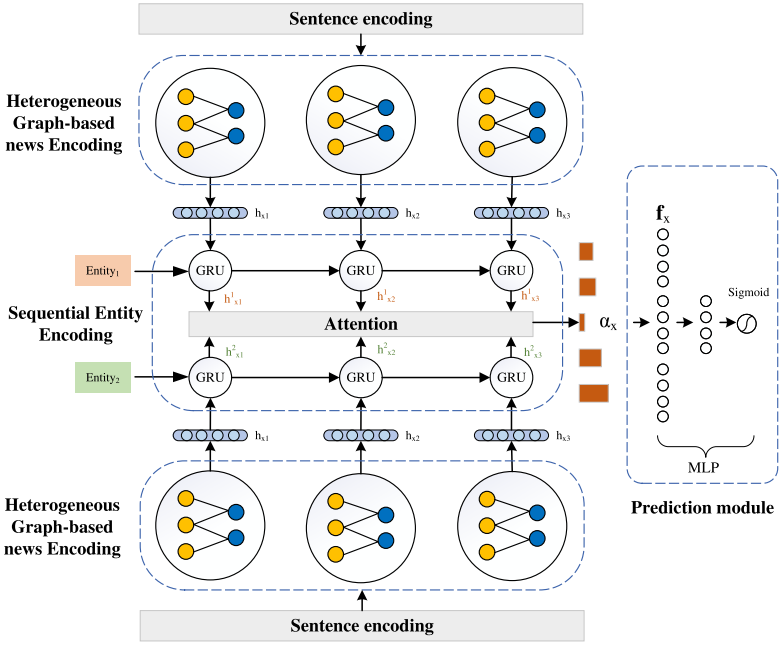

Fig. 2. Framework overview of F^4

short-term factor of popularity prediction. (3) The **sequential entity encoding module** encodes the sequence of news stories of an entity to represent the long-term factor of popularity prediction. (4) The **attention module** captures associations among news stories and entities to represent the global factor of popularity prediction. (5) The **prediction module** makes the final prediction by integrating all factors. We will introduce each module in the following subsections.

3.2 Sentence Encoding

First, each sentence $\mathbf{s}_i$ is represented as a word embedding matrix $X^{\mathbf{s}_i} \in \mathcal{R}^{M(\mathbf{s}_i) \times D_W}$, where $M(\mathbf{s}_i)$ is the length of sentence $\mathbf{s}_i$, and D_W is the dimension size of the word embedding vectors.

As shown in Fig. 3, the input word embedding matrix is first initialized by pre-trained word embedding. Then, it flows through a CNN layer, which adopts convolutional operations to capture the local n-gram features for each sentence $\mathbf{s}_i$. The output of the CNN component then flows through a BiLSTM layer, which captures dependency between sentences.

3.3 Heterogeneous Graph-Based News Encoding

Inspired by heterogeneous graph [25], we construct a heterogeneous graph for each news story, which consists of two types of nodes: word nodes and sentence

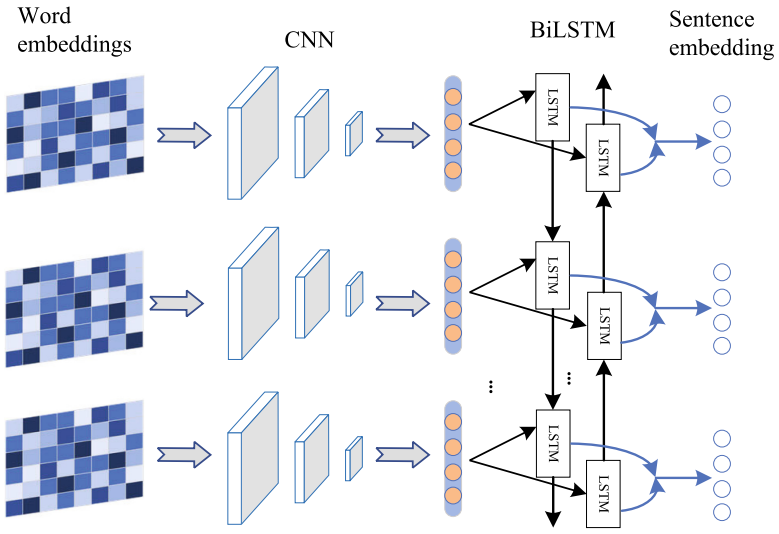

Fig. 3. Sentence encoding

nodes. We initialize each sentence node, i.e., $\mathbf{e}_i^{S,0} \in \mathcal{R}^{D_S}$, for sentence $\mathbf{s}_i$, by the output of sentence encoder in Sect. 3.2. We initialize the word node, i.e., $\mathbf{e}_j^{W,0} \in \mathcal{R}^{D_W}$, for the word $\mathbf{w}_j$, by the pre-trained word embedding vectors. We construct an edge for each pair of word node and sentence node. We initialize the edge vector $\mathbf{e}_{i \leftrightarrow j} \in \mathcal{R}^{D_E}$ for the link between sentence $\mathbf{s}_i$ and word $\mathbf{w}_j$. To distinguish important words from common words, we first compute the TF-IDF weight of each word in each sentence, formally, $tfidf_{i,j} = tf(i,j)/df(j)$, where $tf(i,j)$ is the number of occurrences of word j in sentence i, and $df(j)$ is the number of sentences that contains word j. Then we divide the TF-IDF values to 10 bins, and map each $tfidf(i,j)$ to one vector, corresponding to one bin. The edge vector is not updated during the encoding process.

In the $l-$th layer of graph neural network, we update the hidden states related to sentence nodes by aggregating adjacent word nodes. In order to propagate the popularity information of current buzz words to the corresponding sentence, we first adopt graph attention[24], in Eq. 1:

$$
\begin{aligned}
z_{i,j}^{l+1} &= \text{LeakyReLU}\left(\mathbf{W}_a\left[\mathbf{W}_q\mathbf{e}_i^{S,l} \| \mathbf{W}_k\mathbf{e}_j^{W,l} \| e_{i \leftrightarrow j}\right]\right) \\
\alpha_{i,j}^{l+1} &= \frac{\exp\left(\mathbf{z}_{i,j}^{l+1}\right)}{\sum_{j \in \mathcal{N}_i} \exp\left(\mathbf{z}_{i,j}^{l+1}\right)} \\
\mathbf{u}_i^{l+1} &= \sigma\left(\sum_{j \in \mathcal{N}_\mathbf{s}} \alpha_{i,j}^{l+1}\mathbf{W}_v\mathbf{e}_j^{W,l}\right)
\end{aligned}
\tag{1}
$$

where $\|$ is the concatenation operation, σ is the sigmoid function, $\mathbf{W}_a, \mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$ are trainable weights, $\mathcal{N}_i$ is the set of adjacent word nodes of

sentence $\mathbf{s}_i$, and $\alpha_{i,j}^{l+1}$ is the attention weight between sentence node $\mathbf{s}_i$ to word node $\mathbf{w}_j$. We also adopt multi-head attention. As shown in Eq. 2, suppose $\alpha_{i,j}^{l+1,k}$ denotes the $k-$th attention head,

$$\mathbf{u}_i^{l+1} = \|_{k=1}^{K} \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{i,j}^{k,l+1} \mathbf{W}_v^k \mathbf{e}_i^{W,l} \right) \tag{2}$$

We also add a residual connection to avoid gradient vanishing after several iterations. Therefore, the hidden state of sentence node can be represented as:

$$\mathbf{e}_i^{S,l+1} = \mathbf{u}_i^{l+1} + \mathbf{e}_i^{S,0} \tag{3}$$

We apply similar computation to update the hidden states of word nodes in each iteration. Finally, we apply a position-wise feed-forward (FFN) layer consisting of two linear transformations just as Transformer[23] on all sentences to output the news encoding.

$$\mathbf{h}_{\mathbf{x}} = \mathrm{FFN}\big((\mathbf{e}_1^{S,L} \| \cdots \| \mathbf{e}_{N(\mathbf{x})}^{S,L})\big) \tag{4}$$

where $N(\mathbf{x})$ is the number of sentences in news $\mathbf{x}$, and L is the number of layers in the graph neural network.

3.4 Sequential Entity Encoding

For each news story, we also extract the entities. Suppose for each news story $\mathbf{x}$, the set of entities extracted from $\mathbf{x}$ is $\mathcal{O}(\mathbf{x})$. We construct an *entity sequence* for each entity o, i.e., $c_o =< \mathbf{x}_{o,1}, \cdots, \mathbf{x}_{o,T(o)} >$, which is the chronically ordered sequence of news stories containing the entity, i.e., $o \in O(\mathbf{x}_{o,t})$, where t is the released time of $\mathbf{x}_{o,t}$ and $T(o)$ is the number of news stories which contain entity o.

As shown in Fig. 2, the sequential entity encoding module operates the Gated Recurrent Unit (GRU) upon the entity sequence. We feed the GRU layer with the hidden state by news encoding from Sect. 3.3. Then GRU adopts a reset gate, an update gate, and a current memory gate to update the hidden state of entity from previously released news, as in Eq. 5

$$\begin{aligned} z^{o,t} &= \sigma\left(\mathbf{W}_z \cdot \left[\mathbf{h}_{\mathbf{x}_{o,t}} \| \mathbf{o}^{t-1}\right]\right) \\ r^{o,t} &= \sigma\left(\mathbf{W}_r \cdot \left[\mathbf{h}_{\mathbf{x}_{o,t}} \| \mathbf{o}^{t-1}\right]\right) \\ \tilde{\mathbf{o}^t} &= \tanh\left(\mathbf{W}_o \cdot \left[r^{o,t}\mathbf{o}^{t-1} \| \mathbf{h}_{\mathbf{x}_{o,t-1}}\right]\right) \\ \mathbf{o}^t &= \left(1 - z^{o,t}\right) \times \mathbf{o}^t + z^{o,t-1} \times \tilde{\mathbf{o}^{t-1}} \end{aligned} \tag{5}$$

where $\mathbf{W}_z$, $\mathbf{W}_r$, $\mathbf{W}_o$ are learnable GRU weight matrices, and $\sigma(\cdot), tanh(\cdot)$ are the sigmoid and tanh activation functions, respectively.

3.5 Attention

Since each news story is related to several entities, and each entity is related to numerous news stories, we further adopt an attention layer to capture the global interactions among entities and news stories. Suppose o, p are entities extracted from news story, i.e., $o \in \mathcal{O}(\mathbf{x}), p \in \mathcal{O}(\mathbf{x})$, and o is the entity with the longest sequence, i.e., $o = \arg\max_{T(o)} O(\mathbf{x})$, the attention layer is designed as follows:

$$\begin{aligned} z_{\mathbf{x}}^{o,p} &= \text{LeakyReLU}\big(\mathbf{W}_M \left(\mathbf{W}_q \mathbf{o}^t \| \mathbf{W}_k \mathbf{p}^t\right)\big) \\ \alpha_{\mathbf{x}}^{o,p} &= \frac{\exp\left(z_{\mathbf{x}}^{o,p}\right)}{\sum_{p \in \mathcal{O}(\mathbf{x})} \exp\left(z_{\mathbf{x}}^{o,p}\right)} \\ \mathbf{f}_{\mathbf{x}} &= \sum_{p \in \mathcal{O}(\mathbf{x})} \sigma\big(\alpha_{\mathbf{x}}^{o,p} \mathbf{W}_v \mathbf{o}^t\big) \end{aligned} \tag{6}$$

where $\mathbf{W}_M, \mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$ are trainable weights.

3.6 Prediction

Finally, we can use the news representation obtained in Sect. 3.5 in Eq. 6 to feed a towered MultiLayer Perceptron (MLP) component.

$$\hat{y} = \sigma\Big(f_N\Big(\cdots f_2\big(f_1(\mathbf{f}_{\mathbf{x}})\big)\cdots\Big)\Big) \tag{7}$$

where $f_l(\cdot)$ with $l = 1, 2, \cdots, N$ denotes the mapping function for the l-th hidden layer in MLP. $f_l(\mathbf{x}) = \sigma(\mathbf{W}_l \mathbf{x} + \mathbf{b}_l)$, where $\mathbf{W}_l$ and $\mathbf{b}_l$ are learnable weight matrix and bias vector for layer l. The activation function σ for each layer is sigmoid. We set the size of layers (i.e., the dimensionality of $\mathbf{x}$) as one-third of the previous layers. The output layer $f_{out}(\cdot)$ is similar to $f_l(\cdot)$ and its size is the number of selected items.

The Loss function is Mean Absolute Error(MAE), which is given by:

$$\mathrm{L} = \frac{1}{|\mathcal{D}|} \sum_{<\mathbf{x},y> \in \mathcal{D}} \big(\,|\hat{y} - y|\,\big) \tag{8}$$

4 Experiment

4.1 Dataset

We used three datasets for evaluation. The statistics of the datasets are shown in Table 1.

Disaster. We crawled 50,000 Chinese news related to catastrophic events published by the famous news outlet "toutiao" on the Weibo platform during 2015 to 2020 . We labeled each story (i.e., y equals to the sum of numbers of likes, comments, and thumb-ups) at crawl time (January 2021) on Weibo for each story. Catastrophic events mainly include earthquakes, tsunamis, floods, strong winds, sandstorms, landslides, and typhoons occurred in China.

Table 1. Statistics of datasets

Data	#News	#Words	#Entity
Disaster datasets	50,000	100,000	15,000
Entertainment datasets	48,000	68,000	9,000
MIND datasets	50,000	60,000	12,000

Entertainment. We also crawled 48,000 news, published by "toutiao" on the Weibo platform during the year 2018 to 2020, to build a dataset on entertainment-related topics, including movies, shows, music, celebrities, gossip, and so on.

MIND[1] is a commonly used benchmark dataset in news recommendation. We extracted 50,000 news which have more interactions with users. Then, we label each story by the total number of users who browsed and clicked the news.

4.2 Experimental Setup

Data Pre-Processing. For two Chinese datasets, we used the Chinese named entity recognition tool LAC[2] to extract the factual entities. In the MIND dataset, we retained the content and entity information of each news. In text pre-processing, we removed emoji expressions, HTTP links, and mentions (@somebody) in the news content. For each news, we divided the news into a set of sentences, and we used the jieba Natural Language Processing tool[3] for segmentation. In Chinese datasets, we initialized word embedding vectors with 128-dimensional pre-trained embedding from AI Tencent[4]. In the MIND dataset, we initialized with 300-dimensional GloVe embedding [17]. We filtered stop words. Sentence segmentation was conducted based on punctuation marks ".". The maximum number of sentences in each news story is set to 5. To eliminate the common noise words, we further removed 10% of the vocabulary with low TF-IDF values over the whole dataset. We used random $80\% - 10\% - 10\%$ training/valid/test split. All the codes and data are publically available in GitHub[5].

Parameters. For BiLSTM in sentence encoding, we used 2 layers, with 128−dimensional hidden states and 128−dimensional output sentence encoding. For heterogeneous graph based news encoding, we used 8 attention heads, 50−dimensional edge vector, 64−dimensional hidden states in graph neural network, 512−dimensional hidden states in the 2−layer FFN to output a 50−dimensional news encoding. For sequential entity encoding, we used

[1] https://msnews.github.io/.
[2] https://github.com/baidu/lac.
[3] https://github.com/fxsjy/jieba.
[4] https://ai.tencent.com/ailab/nlp/data/Tencent_AILab_ChineseEmbedding.tar.gz.
[5] https://github.com/XMUDM/NewsPopularityPrediction.

128−dimensional hidden states. For the news popularity prediction module, we set the number of MLP layers to 3. During training, we used a batch size of 256 and apply Adam optimizer [7] with an initial learning rate of 5e–3. We set the decay of learning rate until the lowest was 5e–5, and an early stop was performed when validation loss did not decrease for three continuous epochs.

Baselines. We compared F^4 with the following baselines. (1) **SVM** Support Vector Machine (SVM) was used in [2,21] on traditional bag of stemmed words (BOW) vectors. (2) **MLP+CNN** Similar as [3], we implemented two CNN layers, followed by a dense layer, on a concatenation of W2V vectors and TF-IDF vectors. (3) **LSTM** We learned news representation with LSTM, where for each news, we used a CNN component to extract local sentence encoding vectors to feed a 3-layer LSTM. (4) **BiLSTM** We used a PCNN on words and 3-layer BiLSTM on sentences to get the characteristics of the news. (5) **GCN** was adopted on the word-news graph.

4.3 Comparative Regression Study

To study the accuracy of predicted news popularity, we adopted regression evaluation metrics, including Mean Absolute Error (MAE) in Eq. 8, Root Mean Squared Error (RMSE), and Median Absolute Error(MedAE).

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{D}|}\sum_{<\mathbf{x},y>\in\mathcal{D}}(\hat{y}-y)^2}, \quad \text{MedAE} = median\{|y-\hat{y}|\} \tag{9}$$

where $|\mathcal{D}|$ represents the number of news stories in the dataset, y is the true popularity, $\hat{y}$ is the predicted popularity, $median\{|y-\hat{y}|\}$ returns the median value in the set $\{|y-\hat{y}|\}$.

From Table 2, we have the following observations. (1) F^4 *consistently outperformed all baselines in terms of RMSE, MedAE, and MAE.* (2) F^4 *produced robust regression performance over three datasets.* (3) Besides F^4, GCN was the second-best method in terms of RMSE, while BiLSTM produced the lowest MedAE and MAE. This suggests that adopting a graph neural network or sequential model per se can not guarantee a robust performance.

Table 2. Comparative Regression results of different baselines

Model	Disaster dataset			Entertainment dataset			MIND dataset		
	RMSE	MedAE	MAE	RMSE	MedAE	MAE	RMSE	MedAE	MAE
SVM	0.0821	0.0246	0.0261	0.0184	0.0078	0.0103	0.1654	0.0524	0.1034
MLP+CNN	0.0704	0.0143	0.0235	0.0156	0.0047	0.0071	0.1281	0.0441	0.0725
LSTM	0.0682	0.0112	0.0167	0.0134	0.0018	0.0034	0.1238	0.0256	0.0595
BiLSTM	0.0669	0.0105	0.0159	0.0135	0.0016	0.0056	0.1338	0.0363	0.0699
GCN	0.0581	0.0121	0.0187	0.0131	0.0022	0.0035	0.1254	0.0125	0.0745
F^4	**0.0424**	**0.0083**	**0.0146**	**0.0087**	**0.0006**	**0.0025**	**0.1076**	**0.0021**	**0.0359**

4.4 Comparative Ranking Study

We were also concerned about whether the order of news popularity is predicted precisely, and we evaluated the ranking performance in terms of two evaluation metrics. For each entity, we derived a gold-standard list of the news based on the actual popularity. Then, we computed $HR@K, NDCG@K$ by comparing each method's ranking list of predicted popularity with the gold-standard list:

$$HR@K = \frac{\sum_{j=1}^{K} IK_j}{K}, \quad NDCG@K = \frac{1}{Z_K} \sum_{j=1}^{K} \frac{2^{rel_j} - 1}{\log_2(1+j)}, \tag{10}$$

where K refers to the topK ranking result, IK_j returns 1 if the news story at position j is within the topK most popular stories in the gold-standard list. Z_K is a normalizer which ensures that perfect ranking has a value of 1, rel_j is the relevance score at position j. We set $rel_j = 1$ if the jth result is within the top 5 items in the gold-standard list, and $rel_j = 0$ otherwise.

In Fig. 4, We reported the average HR@5 and NDCG@5 results in three datasets. We have three remarks. (1)F^4 *consistently outperformed all competitors in terms of both HR@5 and NDCG@5.* F^4 did not only retrieve popularity news (i.e., high HR@5) but also distinguished most and more popular news(i.e., high NDCG@5). (2) F^4 *produced robust ranking performance over different datasets.* (3) The performance of GCN is worse than that of the sequential model in the Entertainment dataset and MIND dataset. The underlying reason is that graph-based encoding can not capture the long-term influence of news popularity appropriately. This observation validated our assumption of modeling the entity sequences for news popularity prediction.

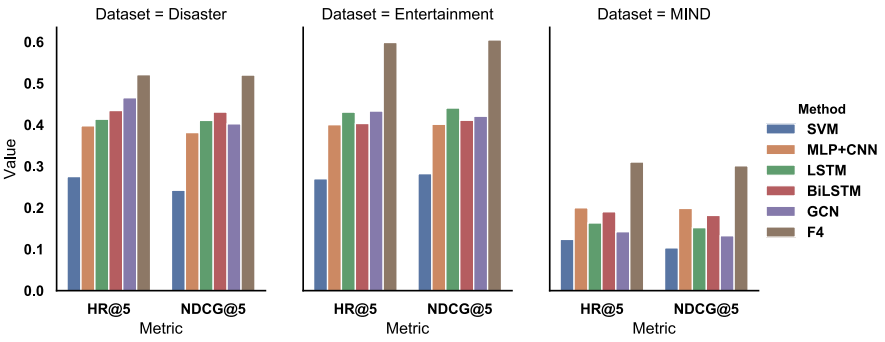

Fig. 4. Comparative ranking performance of different baselines

5 Ablation Study

In order to better understand the contributions of different modules, we conducted an ablation study. (1) F^4 **-entity** removes the sequential entity encoding

module in Sect. 3.4; (2) F^4 **-graph** removes the heterogeneous graph based news encoding module in Sect. 3.3; (3) F^4 **-both** removes both the sequential entity encoding module and the heterogeneous graph based news encoding module.

As shown in Table 3, we have the following observations. (1) F^4-entity and F^4-graph were comparable. F^4-entity produced lower RMSE and MedAE, higher MAE in Disaster and Entertainment Datasets, and higher RMSE and lower MedAE and MAE in MIND dataset. This shows that the sequential entity encoding module and the graph-based news encoding module compensate each other, with the former capturing the long-term popularity factor while the latter capturing the short-term popularity factor. (2) F^4-both performed significantly worse than F^4-entity and F^4-graph in terms of most evaluation metrics. This observation verifies the importance of the sequential entity encoding module and the graph-based news encoding module. (3) Comparing F^4 with the rest baselines, F^4 produced the best results in terms of all evaluation metrics across different datasets.

Table 3. Regression performance of different modules

Model	Disaster dataset			Entertainment dataset			MIND dataset		
	RMSE	MedAE	MAE	RMSE	MedAE	MAE	RMSE	MedAE	MAE
F^4-entity	0.0593	0.0126	0.0139	0.0124	0.0013	0.0029	0.1155	0.0164	0.0564
F^4-graph	0.0512	0.0098	0.0146	0.0105	0.0009	0.0031	0.1228	0.0065	0.0481
F^4-both	0.0651	0.0103	0.0156	0.0152	0.0017	0.0048	0.1272	0.0325	0.0652
F^4	**0.0424**	**0.0083**	**0.0146**	**0.0087**	**0.0006**	**0.0025**	**0.1076**	**0.0021**	**0.0359**

The ranking results were shown in Fig. 5, We have the following remarks. (1) F^4-both performed the worst, while F^4 performed the best, in terms of all ranking evaluation metrics, across different datasets. (2) F^4-entity performed significantly worse than F^4-graph, which shows that capturing the long-term factor in entity sequence is especially important for accurate popularity ranking.

6 Performance of Attention in Merging Entities

Finally, we evaluated the impact of the attention mechanism. We implemented several different baselines to merge news embedding. (1) F^4-nomerge fed the prediction module with the news encoding output by a single entity sequence. First, we computed the TF-IDF weighting of each entity in each news story. Then, we used one entity with the maximal TF-IDF weight to represent each news story. Next, we implemented the sequential entity encoding module in Sect. 3.4 and used the output of the corresponding unit of the representative entity sequence. (2) F^4-avg used average pooling to merge the output of news embedding in several entity sequences. (3) F^4-max used max-pooling to merge

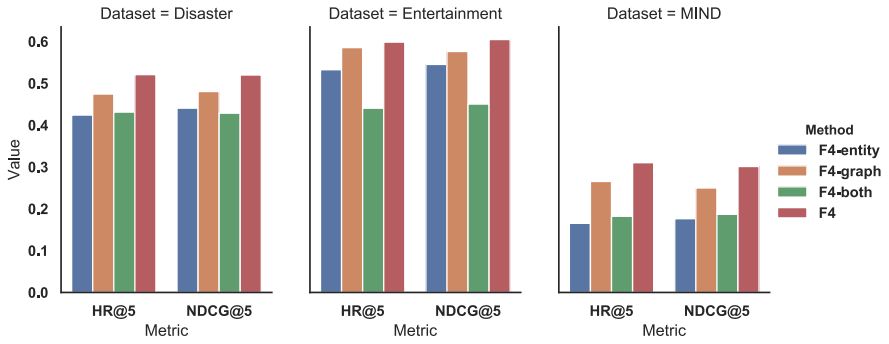

Fig. 5. Ranking performance of different modules

the news embedding. (4) F^4-entity used the pre-trained embedding of entities to calculate attention weights.

As shown in Table 4, F^4 consistently performed the best in terms of regression evaluation metrics. This indicates the effectiveness of the attention mechanism used in merging the news embedding from various relevant entities. In addition, F^4-entity generated the second-best RMSE results in the Disaster and MIND dataset. However, the performance was not stable. One possible reason is that F^4-entity calculated the attention weights based on static entity profiles and did not reflect the dynamic nature of entity sequence.

Table 4. Regression performance of different merging strategies

Model	Disaster dataset			Entertainment dataset			MIND dataset		
	RMSE	MedAE	MAE	RMSE	MedAE	MAE	RMSE	MedAE	MAE
F^4-nomerge	0.0451	0.0142	0.0149	0.0106	0.0009	0.0025	0.1134	0.0152	0.0521
F^4-avg	0.0438	0.0122	0.0149	0.0112	0.0012	0.0031	0.1326	0.0084	0.0674
F^4-max	0.0539	0.0144	0.0191	0.0091	0.0010	0.0028	0.1259	0.0249	0.0612
F^4-entity	0.0445	0.0095	0.0167	0.010	0.0011	0.0027	0.1103	0.0095	0.0498
F^4	**0.0424**	**0.0083**	**0.0146**	**0.0087**	**0.0006**	**0.0025**	**0.1076**	**0.0021**	**0.0359**

The ranking results are shown in Fig. 6. Again, F^4 consistently gained significant performance improvements in terms of both HR@5 and NDCG@5. Interestingly, the ranking performance of F^4-nomerge was better than naive merge, e.g., either F^4-avg or F^4-max. The underlying reason is that, though many entities are involved in a news event, they play different roles, and how to integrate the different entities in predicting news popularity is a non-trivial task. Many factors must be considered, including the relevance of the entity, the temporal impact of the entity, and correlations among entities.

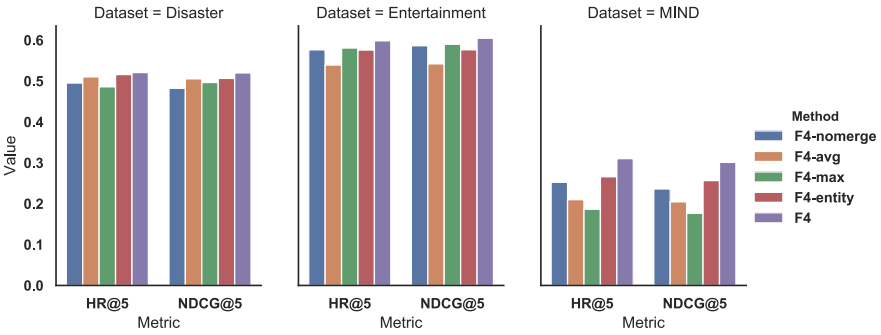

Fig. 6. Ranking performance of different merging strategies

7 Conclusion

In this paper, we study the problem of news popularity prediction. We discuss the local, global, short-term, and long-term factors that affect news popularity. We present F^4, which adopts a CNN based sentence encoding module to represent the local context of each news story; a heterogeneous graph-based module to capture the short-term information propagation from current buzz words to each news story; a sequential module to extract long-term popularity features in entity sequence; and finally, an attention module to learn global news-entity correlations. Experiments on real-world English and Chinese datasets have demonstrated the effectiveness of F^4. In the future, we plan to investigate the problem of forecasting public response to news events via multi-tasking, e.g., incorporating popularity prediction, sentiment classification, and so on.

Acknowledgements. Chen Lin is the corresponding author. Chen Lin is supported by the Natural Science Foundation of China (No. 61972328), Joint Innovation Research Program of Fujian Province China (No.2020R0130). Hui Li is supported by the Natural Science Foundation of China (No. 62002303), Natural Science Foundation of Fujian Province China (No. 2020J05001). Quan Zou is supported by Natural Science Foundation of China (No. 61922020).

References

1. Yang, Y., Liu, Y., Lu, X., Xu, J., Wang, F.: A named entity topic model for news popularity prediction. Knowl.-Based Syst. **208**, 106430 (2020)
2. Ambroselli, C., Risch, J., Krestel, R., Loos, A.: Prediction for the newsroom: Which articles will get the most comments? In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans - Louisiana, vol. 3 (Industry Papers), pp. 193–199 (2018)
3. Davoudi, H., An, A., Edall, G.: Content-based dwell time engagement prediction model for news articles. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, vol. 2 (Industry Papers), pp. 226–233 (2019)

4. Gupta, R.K., Yang, Y.: Predicting and understanding news social popularity with emotional salience features. In: Proceedings of the 27th ACM International Conference on Multimedia, New York, NY, USA, pp. 139–147 (2019)
5. Hamid, A., et al.: Fake news detection in social media using graph neural networks and NLP techniques: a COVID-19 use-case. In: MediaEval. CEUR Workshop Proceedings, vol. 2882 (2020)
6. Keneshloo, Y., Wang, S., Han, E.S., Ramakrishnan, N.: Predicting the popularity of news articles. In: Venkatasubramanian, S.C., Jr., W.M. (eds.) Proceedings of the 2016 SIAM International Conference on Data Mining, Miami, Florida, USA, 5–7 May 2016, pp. 441–449 (2016)
7. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (Poster) (2015)
8. Lee, J.G., Moon, S.B., Salamatian, K.: An approach to model and predict the popularity of online contents with explanatory factors. In: Web Intelligence, pp. 623–630 (2010)
9. Lee, J.G., Moon, S.B., Salamatian, K.: Modeling and predicting the popularity of online contents with cox proportional hazard regression model. Neurocomputing **76**(1), 134–145 (2012)
10. Lin, P., Mo, X., Lin, G., Ling, L., Wei, T., Luo, W.: A news-driven recurrent neural network for market volatility prediction. In: ACPR, pp. 776–781 (2017)
11. Liu, Q., Cheng, X., Su, S., Zhu, S.: Hierarchical complementary attention network for predicting stock price movements with news. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18, pp. 1603–1606. Association for Computing Machinery, New York (2018)
12. Mazloom, M., Rietveld, R., Rudinac, S., Worring, M., van Dolen, W.: Multimodal popularity prediction of brand-related social media posts. In: Proceedings of the 24th ACM International Conference on Multimedia, MM '16, New York, NY, USA, pp. 197–201 (2016)
13. Mei, H., Eisner, J.: The neural hawkes process: a neurally self-modulating multivariate point process. In: NIPS, pp. 6754–6764 (2017)
14. Mukherjee, S.K., Bandyopadhyay, S.: Clustering to determine predictive model for news reports analysis and econometric modeling. In: ReTIS, pp. 302–309 (2015)
15. Naseri, M., Zamani, H.: Analyzing and predicting news popularity in an instant messaging service. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'19, New York, NY, USA, pp. 1053–1056 (2019)
16. Okano, E.Y., Liu, Z., Ji, D., Ruiz, E.E.S.: Fake news detection on fake.br using hierarchical attention networks. In: Quaresma, P., Vieira, R., Aluísio, S., Moniz, H., Batista, F., Gonçalves, T. (eds.) PROPOR 2020. LNCS (LNAI), vol. 12037, pp. 143–152. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-41505-1_14
17. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP, pp. 1532–1543. ACL (2014)
18. Sadiq, S., Wagner, N., Shyu, M., Feaster, D.: High dimensional latent space variational autoencoders for fake news detection. In: MIPR, pp. 437–442 (2019)
19. Shang, Y., Wang, Y.: Study of CNN-based news-driven stock price movement prediction in the a-share market. In: Qin, P., Wang, H., Sun, G., Lu, Z. (eds.) ICPCSEE 2020. CCIS, vol. 1258, pp. 467–474. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-7984-4_35
20. Stoddard, G.: Popularity dynamics and intrinsic quality in reddit and hacker news. In: ICWSM, pp. 416–425 (2015)

21. Tsagkias, M., Weerkamp, W., de Rijke, M.: Predicting the volume of comments on online news stories. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09, New York, NY, USA, pp. 1765–1768 (2009)
22. Tsagkias, M., Weerkamp, W., de Rijke, M.: Predicting the volume of comments on online news stories. In: CIKM, pp. 1765–1768 (2009)
23. Vaswani, A., et al.: Attention is all you need. In: NIPS, pp. 5998–6008 (2017)
24. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks (2017). CoRR arXiv:1710.10903
25. Wang, D., Liu, P., Zheng, Y., Qiu, X., Huang, X.: Heterogeneous graph neural networks for extractive document summarization. In: ACL, pp. 6209–6219 (2020)
26. Wu, C., Wu, F., An, M., Huang, J., Huang, Y., Xie, X.: NPA: neural news recommendation with personalized attention. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '19, pp. 2576–2584. Association for Computing Machinery, New York (2019)
27. Wu, C., Wu, F., An, M., Huang, Y., Xie, X.: Neural news recommendation with topic-aware news representation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 1154–1159 (2019)
28. Zaman, T., Fox, E.B., Bradlow, E.T.: A bayesian approach for predicting the popularity of tweets (2013). CoRR arXiv:1304.6777
29. Zhao, X., Wang, C., Yang, Z., Zhang, Y., Yuan, X.: Online news emotion prediction with bidirectional LSTM. In: Cui, B., Zhang, N., Xu, J., Lian, X., Liu, D. (eds.) WAIM 2016. LNCS, vol. 9659, pp. 238–250. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39958-4_19