Popularity Prediction on Online Articles with Deep Fusion of Temporal Process and Content Features

Dongliang Liao,[1] Jin Xu,[1*] Gongfu Li,[1] Weijie Huang,[1] Weiqing Liu,[2] Jing Li[2]
[1] WeChat, Tencent Inc.
{brightliao, jinxxu, gongfuli,wainhuang}@tencent.com;
[2] University of Science and Technology of China,
cslwqxx@mail.ustc.edu.cn,lj@ustc.edu.cn

Abstract

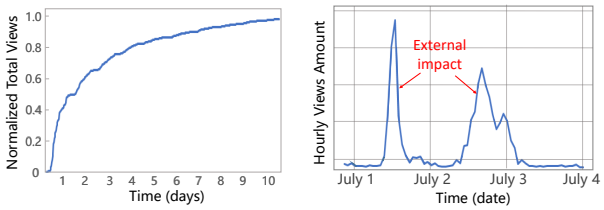
Predicting the popularity of online article sheds light to many applications such as recommendation, advertising and information retrieval. However, there are several technical challenges to be addressed for developing the best of predictive capability. (1) The popularity fluctuates under impacts of external factors, which are unpredictable and hard to capture. (2) Content and meta-data features, largely determining the online content popularity, are usually multi-modal and non-trivial to model. (3) Besides, it also needs to figure out how to integrate temporal process and content features modeling for popularity prediction in different lifecycle stages of online articles. In this paper, we propose a Deep Fusion of Temporal process and Content features (DFTC) method to tackle them. For modeling the temporal popularity process, we adopt the recurrent neural network and convolutional neural network. For multi-modal content features, we exploit the hierarchical attention network and embedding technique. Finally, a temporal attention fusion is employed for dynamically integrating all these parts. Using datasets collected from WeChat, we show that the proposed model significantly outperforms state-of-the-art approaches on popularity prediction.

Introduction

Online articles, such as news in portal websites and blogs in social networks, have become the most important source of information. The popularity of online article describes how much attention it receives, which could be measured by the amount of total views. Popularity is a measure of content quality for content providers, and a way to filter information for content consumers. Unfortunately, we can only acquire the overall popularity after the lifecyle of the online article. Predicting the overall popularity in early stage sheds light to many applications, such as recommendation, advertising and information retrieval (Gao et al. 2018; Liu et al. 2016). Besides, it is also academic valuable and industrial applicable to properly answer the question like "How to predict the overall popularity of online content at any time?"

Recently, popularity prediction has drawn great attentions. Scholars handle this task with two-broad-category approaches: temporal process modeling and content feature modeling. Temporal process modeling predicts popularity based on temporal evolution processes of aggregated view volumes in time slots. The accumulative popularity increases over time while shows unexpected outbreaks under impacts of external factors, shown in Fig. 1. Most existing works capture short term fluctuations based on the specific assumption about external impact (Zhao et al. 2015; Cao et al. 2017; Rizoiu et al. 2017). However, many external factors are unpredictable. Specific assumptions restrict these models' predictive power. Yu *et al.* tried to extract popularity fluctuations from temporal process itself, based on handcrafted "phases" (Yu et al. 2015). However, influences of external factors may cover different ranges and durations. It is hard to assume the amount and shapes of fluctuations artificially. How to extract short term fluctuations automatically is still a unsolved problem in this branch of methods.

On the other hand, recent works have proved the effectiveness of content features in the popularity prediction, such as short text descriptions, titles and images (Zhang et al. 2018; Piotrkowicz et al. 2017; Sanjo and Katsurai 2017). However, online articles are usually long texts which are non-trivial to model, and diverse forms of meta-data features further complicate the content feature modeling. None of existing works has taken fully advantages of the long text and meta-data features for popularity prediction of online articles.



(a) Long term growth trend. (b) Short term fluctuation.

Figure 1: These two sub-figures show temporal dynamic of articles popularity in WeChat[1]. (a) The normalized total views accumulative process of all articles. (b) The hourly views amount variation of an example article.

[1] http://www.wechat.com

*Corresponding author: Jin Xu <jinxxu@tencent.com>; the first two authors contributed equally

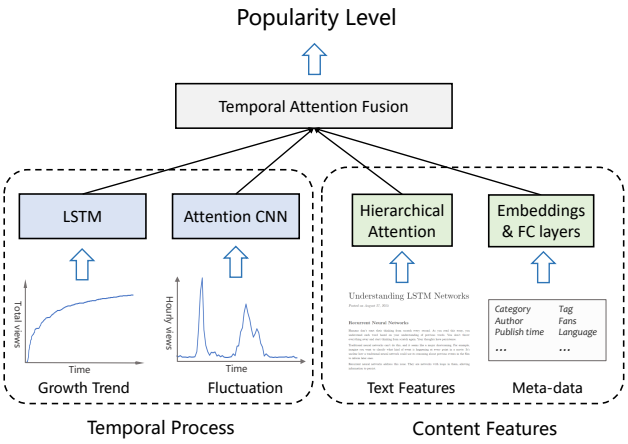


Figure 2: The overall framework of Deep Fusion of Temporal process and Content features model.

Meanwhile, these two categories of methods have their own strength and drawbacks over different lifecycle stages of online article. Temporal process modeling relies on series of history events, and performs better and better over time since the observed popularity gets closer to the overall popularity. However, it is hard to learn the overall trend of popularity at the very beginning after online content have been published. In practical applications, it is valuable to predict the overall popularity in early stage of online article's lifecycle, so that we can recommend potential "hot" articles and filter arid ones. In contrast, content features will not change over time. Thus content features modeling is more reliable in early stage, while it fails to exploit the temporal evolution of popularity. Therefore, we should integrate the temporal process and content feature modeling to leverage their respective power. Nevertheless, different articles show different increase rates and fluctuations in popularity evolutions. Intuitive fusion methods, such as vector concatenation or linear combination, lack flexibility for handling the diversity of popularity evolution processes.

Motivated by above challenges, we propose a neural network method named Deep Fusion of Temporal process and Content features (DFTC). The framework of DFTC is shown in Fig. 2. In our model, we tackle above challenges with following modeling techniques: (1) For modeling the temporal process, we adopt Recurrent Neural Network (RNN) to capture the long term growth trend of popularity. As for short-term fluctuations, we adopt attention based Convolutional Neural Network (CNN) to extract rising or falling "phases" structures automatically. (2) For modeling content features, we exploit Hierarchical Attention Network (HAN) (Yang et al. 2016) for capturing text features and employ embedding techniques to embed meta-data features into homologous dense vectors. (3) For dynamic fusion, we employ a temporal attention layer. It leverages attention mechanism to learn flexible weights for combining all above modeling techniques, based on their outputs and temporal contexts. Besides, we collect real-world datasets from WeChat, and conduct extensive experiments for evaluating prediction performances in different stages, including a case study for effects of CNN and attention fusion.

Our main contributions are summarized as follows:

- We leverage RNN for long term growth trend and CNN for short term fluctuation of temporal processes automatically, rather than specific assumption of external factor or hand-crafted "phases".
- We adopt HAN for text features, embedding techniques for meta-data features, and temporal attention fusion for integrating temporal process and content feature modeling dynamically.
- Experimental results show the proposed model significantly outperforms state-of-the-art methods, demonstrating the validity and superiority of our approach.

Related Work

Popularity prediction has drawn great attentions for decades. Scholars handled this task with two-broad-category approaches: temporal processes modeling and feature-based modeling.

Temporal process modeling. Some researchers regarded the popularity cumulation of online content as an micro arrival point process of view events. They predicted popularity by modeling micro point processes of single events based on reinforced possion processes (Shen et al. 2014), hawkes point processes (Zhao et al. 2015) or neural network (Cao et al. 2017; Gou et al. 2018). However, the amount of events can be explode in a short time in large-scale applications, which will cause performance issues of micro temporal process modeling. Thus we argue that predicting popularity based on macro accumulation process of event volumes has more practical value, since macro temporal processes are typically a few dozens to a few hundred data points.

A number of models have been proposed to describe the evolution of macro temporal process. Hawkes intensity processes (Rizoiu et al. 2017) expanded hawkes point process for macro temporal process and adopted it in Youtube video popularity prediction. HIP made specific assumptions about functional forms of temporal processes and influences of external factors, which may restrict the expressive power of these models (Du et al. 2016). Mishra *et al*. proposed a dual RNN model for modeling both micro and macro temporal process (Mishra, Rizoiu, and Xie 2018) and achieved state-of-the-art performance. However, it still needs micro events for modeling the fluctuation caused by external influence. Yu *et al*. extracted rising and falling "phases" from macro temporal process to capture fluctuations, and proposed a phase-based linear regression approach for popularity prediction (Yu et al. 2015). However, hand-crafted "phases" can not handle the diversity of popularity evolution processes. In our model, we adopt attention CNN for extracting local rising and falling structures automatically and LSTM for capturing the long term growth trend of macro temporal processes.

Feature-based modeling. Some other researchers modeled content and meta-data features of online content for

popularity prediction. Piotrkowicz *et al*. predicted popularity of news articles using only headline features (Piotrkowicz et al. 2017). Sanjo *et al*. proposed a visual-semantic fusion model for online recipe popularity prediction, leveraging image and short text features in recipes (Sanjo and Katsurai 2017). User-guided hierarchical attention network (Zhang et al. 2018) learned modalities content and user features for social image popularity prediction. Unfortunately, there is none existing work has taken fully advantages of the long text and meta-data features for popularity prediction of online articles. Besides, these approaches also ignore popularity evolution processes of online content.

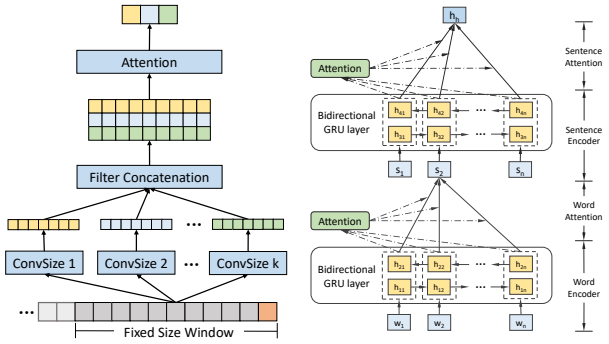
Some other methods extracted various hand-crafted features for popularity prediction, including both temporal process and content features. Keneshloo *et al*. extracted meta-data, content and temporal features, and adopted tree regression for news popularity prediction (Keneshloo et al. 2016). Shulman *et al*. added novel social structure and early adopter features for improving the prediction performance (Shulman, Sharma, and Cosley 2016). Their performances heavily depend on extracted features. However, these features are hard to design and measure, and are often binded to specific datasets or applications. Inspired the huge success of deep learning, we leverage neural network for modeling both the content features and temporal processes, avoiding laborious feature engineering (Xu et al. 2012; Xu et al. 2017).

Problem Formulation

We regard the popularity prediction task as a classification problem, discretizing the amount of total views to n intervals $\{l_1, l_2, \cdots, l_n\}$ to represent popularity levels of online articles. Our goal is to predict the popularity level at any time after the online article has been published. For the ease of calculation, we discretize continuous time to time slots, and aggregate user feedback events volumes as macro time series. Here, user feedback events contain not only "view" but also "share", "comment" or "like" in many applications. We take all the volumes of these events in time slots t as feedback vector $\boldsymbol{v}_t$. More formally, for a online content c, given any time slot t and the history feedback series $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_t\}$, the objective is to predict the overall popularity level of c.

Model

In this section, we introduce the proposed Deep Fusion of Temporal process and Content features (DFTC) model. The overall framework is presented in Fig. 2. DFTC consists of three parts: temporal process modeling, content feature modeling and attentive fusion. The temporal process modeling takes the history feedback series $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_t\}$ as inputs, and adopting recurrent neural network for modeling the long term growth trend and convolutional neural network for capturing short term fluctuations. In the content features modeling, we leverage hierarchical attention network for learning text features, and embedding technique for extracting meta-data features. At last, we dynamically integrate all these parts through temporal attention fusion.



(a) Attention CNN. (b) HAN

Figure 3: (a) The architecture of attention CNN for capturing short term fluctuations. (b) Hierarchical attention network (HAN) for modeling text content features.

Temporal Process Modeling

In this work, we employ Recurrent Neural Network (RNN) for modeling the temporal evolution process of popularity. Long Short Term Memory (LSTM) is the most widely used RNN structure. We reiterate the formulation of LSTM:

$$i_t = \sigma(W^i x_{t-1} + U^i c_{t-1} + V^i h_{t-1} + b^i) \quad (1)$$

$$f_t = \sigma(W^f x^S_{t-1} + U^f c_{t-1} + V^f h_{t-1} + b^f) \quad (2)$$

$$c_t = f_t * h_{t-1} + i_t * tanh(W^c x_{t-1} + V^c h_{t-1} + b^c) \quad (3)$$

$$o_t = \sigma(W^o x^S_{t-1} + U^o c_{t-1} + V^o h_{t-1} + b^o) \quad (4)$$

$$h_t = o_t * tanh(c_t) \quad (5)$$

The superiority of RNN for temporal modeling is that the hidden state h_t involves all the history information so that we need not to make a specific assumption about functional forms of the history trend (Du et al. 2016). What's more, the memory cell c_t in LSTM ensures the long term dependency can also be captured. Thus we adopt LSTM to learn the long term growth trend of popularity. Concretely, we feed the feedback vector $\boldsymbol{v}$ of each time slot into LSTM, and obtain the history growth pattern in the output vector $\boldsymbol{h}^r_t$.

On the other hand, the short term popularity of online content can be effected by external events and shows unexpected outbreaks (Zhao et al. 2015; Rizoiu et al. 2017; Cao et al. 2017). However, it is very difficult to figure out all impact factors since many of them are unpredictable. Therefore, we suppose to capture the short term fluctuation from the temporal process itself. Considering the short term popularity curve in one dimensional time axis, the fluctuation caused by external factors makes the curve consist of rising and falling phases, which look like "mountains" and "valleys", as shown in Fig. 1b. These "mountains" and "valleys" are translation invariant local structures. Thus we exploit 1-D convolutional neural network, which has been proved optimum for capturing such structures. Furthermore, effects of different factors continue over different time spans, which means "mountains" have different widths. Inspired by the inception module (Szegedy et al. 2016), we adopt multiple

kernels with different sizes to capture different scale of fluctuations, shown in Fig. 3a. After then, we stack outputs of all convolutional kernels vertically.

Note that CNNs usually need fixed size inputs. Thus, we take a clipped series $\{\boldsymbol{v}_{t-k+1}, \boldsymbol{v}_{t-k+2}, \cdots, \boldsymbol{v}_t\}$ with fixed length k before t. Then we apply same padding and get an output series $\{\boldsymbol{c}_{t-k+1}, \boldsymbol{c}_{t-k+2}, \cdots, \boldsymbol{c}_t\}$ with length k too, which captures the fluctuation pattern of the recent history. At last, we need to merge output series through temporal dimension into output vector $\boldsymbol{h}_t^c$. There are several widely used methods for the merging operator, such as vector concatenation, max/mean pooling and linear combination. Here we adopt the attention mechanism (Vaswani et al. 2017) for merging $\{\boldsymbol{c}_{t-k+1}, \boldsymbol{c}_{t-k+2}, \cdots, \boldsymbol{c}_t\}$. Attention mechanism helps the output $\boldsymbol{h}_t^c$ to focus on such time slots that are influenced by external factors, via multiplying different attentive weights α^c to different vectors $\boldsymbol{c}$. The calculation of attentive weights α^c and the output vector $\boldsymbol{h}_t^c$ is as follows:

$$a_i^c = V_i^c \tanh(\sum_{i=1}^{k} W_j^c \boldsymbol{c}_{t-k+i} + \boldsymbol{b}^c) \tag{6}$$

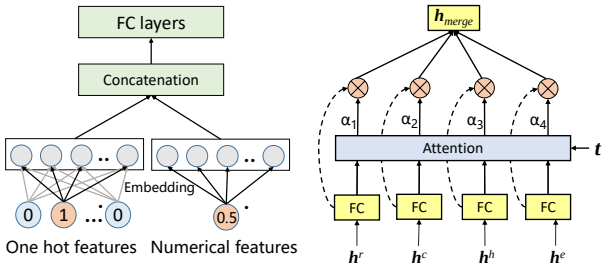
$$\alpha_i^c = \frac{\exp(a_i^c)}{\sum_{j=1}^{k} \exp(a_j^c)} \tag{7}$$

$$\boldsymbol{h}_t^c = \sum_{i=1}^{k} \alpha_i^c \boldsymbol{c}_{t-k+i} \tag{8}$$

Content Features Modeling

Content features of online articles, including text and meta-data features, largely determine their popularity. Online articles are usually long text documents, such as news articles and blogs. Inspired by the huge success of neural network in nature language process (Lai et al. 2015; Yang et al. 2016; Huo, Li, and Zhou 2016), we adopt the Hierarchical Attention Network (HAN) (Yang et al. 2016) for modeling the text content feature. The framework of HAN is showed in Fig. 3b. Considering the inherent hierarchical structure of documents (i.e. words form sentences and sentences form a document), HAN encodes a document to a vector with two levels of encoder and attention, applied at the word-level and sentence-level. Both word-level and sentence-level encoders are bidirectional Gated Recurrent Unit. For more details of HAN, please refer to their research article (Yang et al. 2016). Besides, titles are high-level overviews of articles and show primary impressions. We also learn a title representation vector as a supplement. Since the title is usually a phrase or a sentence, we encode the short text to a vector with only word-level encoder and attention. Then we concatenate document vector and title vector together as text features $\boldsymbol{h}^h$.

Meta-data features consist of both one-hot features, such as category, and numerical features, such as fans number of author. Instead of hand-crafted selection and combination of these features, we exploit embedding techniques for embedding these features into homologous dense vectors and apply fully connected layers for the feature combination.



(a) Embedding. (b) Temporal Attention Fusion.

Figure 4: (a) The illustration of embedding techniques for meta-data features. (b) The architecture of the temporal attention layer.

As shown in Fig. 4a, we embed one-hot features to vectors through embedding matrices. On the other hand, we multiply numerical features by embedding vectors for mapping them to homologous dense vectors. Then we concatenate embedding vectors and apply fully connected layers for combining all meta-data features to $\boldsymbol{h}^e$.

Attentive Fusion

For fusing above modeling techniques, a direct way is to concatenate outputs of all these parts and feed the concatenation result into output layers for prediction. Let $\boldsymbol{h}_t^r, \boldsymbol{h}_t^c, \boldsymbol{h}^h, \boldsymbol{h}^e$ represent outputs of RNN, CNN, HAN and meta-feature embedding. Then we can get prediction at t as: $\hat{y}_t = f(W[\boldsymbol{h}_t^r, \boldsymbol{h}_t^c, \boldsymbol{h}^h, \boldsymbol{h}^e] + \boldsymbol{b})$. In this way, $\boldsymbol{h}_t^r, \boldsymbol{h}_t^c, \boldsymbol{h}^h, \boldsymbol{h}^e$ are combined with fixed weights W. As mentioned in Section 1, we argue that it lacks flexibility for handling the dynamic evolution of temporal process. At the very beginning after online articles have been published, it is hard for temporal process modeling to learn the overall growth trend of popularity. Thus the prediction should mainly depend on content feature modeling. As time goes on, the observed popularity gets closer to the overall popularity, so temporal modeling should take a major part in prediction. On that basis, we suppose to integrate $\boldsymbol{h}_t^r, \boldsymbol{h}_t^c, \boldsymbol{h}^h, \boldsymbol{h}^e$ with a flexible weight $\boldsymbol{\alpha}$. $\boldsymbol{\alpha}$ should be a function of of $\boldsymbol{h}_t^r, \boldsymbol{h}_t^c, \boldsymbol{h}^h, \boldsymbol{h}^e$ and temporal context $\boldsymbol{t}$, so that it could automatically adapt different outputs and temporal context.

In this work, we adopt an attention mechanism to achieve dynamic integration, shown in Fig. 4b. Attention mechanism is a element-wise combination, thus we feed $\boldsymbol{h}_t^r, \boldsymbol{h}_t^c, \boldsymbol{h}^h, \boldsymbol{h}^e$ into fully connected layers for feature combination and acquire element-wise aligned vectors $\hat{\boldsymbol{h}}_t^r, \hat{\boldsymbol{h}}_t^c, \hat{\boldsymbol{h}}^h, \hat{\boldsymbol{h}}^e$. Then we use a two-layer neural network to compute the attentive weights $\boldsymbol{\alpha}^m$ as:

$$a_i^m = V_i^m \tanh(\sum_{j \in \{r,c,h,e\}} W_j^m \hat{\boldsymbol{h}}_t^j + W_t^m \boldsymbol{t} + \boldsymbol{b}^m) \tag{9}$$

$$\alpha_i^m = \frac{\exp(a_i^m)}{\sum_{k \in \{r,c,h,e\}} \exp(a_k^m)} \tag{10}$$

Temporal context $\boldsymbol{t}$ consists of periodic properties (i.e. hour of day and day of week) of the given time slot t, time

interval of t and the publish time. Here, periodic properties are one hot features and time interval is a numerical feature. We apply the same strategy as embedding meta-data features to embed temporal context into vector. With attentive weights $\boldsymbol{\alpha}^m$, we combine all subnetworks dynamically as $\boldsymbol{h}_t^{merge}$ and get a probability distribution $P_t = \{p_t(l_1), p_t(l_2), \cdots, p_t(l_n)\}$ of popularity levels after fully connected layers and softmax output layer. Then we take the popularity level with max probability as prediction result $\hat{y}_t$.

$$\boldsymbol{h}_t^{merge} = \sum_{i \in \{r,c,h,e\}} \alpha_i^m \hat{\boldsymbol{h}}_t^i \tag{11}$$

$$P_t = \text{softmax}(f(\boldsymbol{h}_t^{merge})) \tag{12}$$

$$\hat{y}_t = \arg\max_l p_t(l) \tag{13}$$

Temporal Decayed Loss

With the proposed model, we get a probability distribution P_t of popularity levels of online article c at time slot t. Supposing the true level of c is l_c, the single step loss at t is defined by the cross entropy as $L_t = -\log p_t(l_c)$. Through the whole time series of c, we can define the overall loss of our model as $J = \sum_t L_t$.

In practical applications, it is much more valuable to predict overall popularity in early stage. Besides, the inherent relation between the observed popularity and overall popularity makes it easier to make predictions in the later period. In order to help our model to invest more efforts to optimize the prediction performance in early stage, we multiply a temporal decayed factor to the single step loss:

$$J = \sum_t D(\Delta t) L_t = -\sum_t D(\Delta t) \log p_t(l_c) \tag{14}$$

The temporal decayed factor $D(\Delta t)$ should be a monotonous and non increasing function of the time interval Δt between t and the publish time. In this work, we choose a function as follows:

$$D(\Delta t) = \lceil \log_\gamma(\Delta t + 1) \rceil^{-1} \tag{15}$$

Here, $\lceil \cdot \rceil$ represents up rounding operator. Δt is number of time slots from the publish time to t, thus Δt and $\lceil \log_\gamma(\Delta t + 1) \rceil$ are both positive integers. $\gamma > 1$ is a hyper-parameter for controlling the decay rate. We adopt the log function to ensure the decay rate of $D(\Delta t)$ will get smaller and smaller with time goes by. The up rounding operator is employed for restricting the initial decay rate of log function.

Experiments

Datasets

We collect an online article dataset from a widely used mobile social application WeChat[2]. Both media organizations and personal users can set up their official accounts for publishing news and articles. Users can follow official accounts to subscribe article updates. WeChat provides article recommendation and search function for users. When reading articles, users can also take "share", "save", "like" and "tip" actions. Nowadays, there are more than 500,000 new articles and 2 billion "views" per day in WeChat.

[2]http://www.wechat.com

	hot	normal		cold	
		>1,000	≤1,000	>10	≤10
training #articles	18,832	9,159	9,243	8,946	8,884
balanced test #articles	2,093	1,020	989	1,009	1,007
random test #articles	78	467	2,060	4,308	23,087

Table 1: Datasets Statistics

We divide the overall popularity of articles into three categories, "hot" (more than 10,000 views), "cold" (less than 100 views) and "normal" (otherwise). Here, we take the total views of 15 days after articles are published as an approximation of overall popularity. The distribution of article views is a typical power-law distribution. Only 0.08% articles are "hot", while more than 93% articles are "cold". Thus we collect all "hot" articles from May 25 to July 25 and under sample other two category articles. In order to ensure the diversity of training data, we adopt a piecewise uniform sampling over the logarithmic views amount. We count the amount of "view" ,"share", "save", "like" and "tip" actions per 5 minutes of each article as macro time series. Then we clip these time series before the observed popularity achieves 80% of overall popularity or the 80% of "hot" threshold. The articles whose time series lengths are less than 12 is filtered, because they either become "hot" immediately or retain "cold" all the time. Finally, we get 61,178 articles from WeChat. In following experiments, we take 85% of articles for training our model, 5% articles for validation and 10% articles for evaluation, called *balanced test set*. For evaluating our model on the realistic distribution, we randomly sample other 30,000 articles as *random test set* from July 26 to August 10. The dataset statistics information is shown in Table 1. Besides, meta-data features consist of *"category", "publish time", "content length", "video number"* and *"fans numbers of publisher"*.

Experiment Settings

Baselines We compare the proposed DFTC method with following baselines:

- **Feature-based classifier**. We adopt logistic regression (LR) and random forests (RF) as baselines of the popularity classification task. These classifiers take both temporal process features and content features as inputs.
- **HIP (Rizoiu et al. 2017)**. Hawkes Intensity Process extents the well known hawkes point process for modeling macro temporal process and is applied for predicting popularity of videos.
- **VoRNN-TS (Mishra, Rizoiu, and Xie 2018)**. Volumn

Method	Results of Balanced Test Set				Results of Random Test Set			
	Accuracy	hot F1	normal F1	cold F1	Accuracy	hot F1	normal F1	cold F1
LR	0.6441	0.3575	0.6446	0.7088	0.7551	0.4248	0.8272	0.8973
RF	0.6587	0.4246	0.6506	0.7277	0.8086	0.4743	0.8454	0.8909
HIP	0.6502	0.4353	0.6330	0.7182	0.7860	0.4342	0.7742	0.9217
VoRNN-TS	0.6709	0.4447	0.6530	0.7366	0.8569	0.4581	0.8505	0.9540
CACNN	0.6965	0.4018	0.7040	0.7394	0.8498	0.4825	0.8472	0.9493
DFTC-TS	0.7278	0.4858	0.7203	0.7638	0.8863	0.5253	0.8592	0.9698
DFTC-SF	0.6542	0.5343	0.6754	0.6212	0.6879	0.5536	0.6926	0.7869
DFTC-SM	0.7559	0.5554	0.7489	0.7812	0.9301	0.5649	0.8625	0.9759
DFTC	**0.8147**	**0.6110**	**0.7822**	**0.8393**	**0.9653**	**0.6292**	**0.8729**	**0.9916**

Table 2: Overall Prediction Performance

RNN achieves the state-of-the-art performance on macro temporal proceses, leveraging the superiority of LSTM.

- **CACNN (Gao et al. 2018)**. Context Attention Convolutional Neural Network is proposed for click-through rate forecasting, which models temporal process with attention CNN and incorporate meta-data features.
- **DFTC-TS & DFTC-SF**. DFTC-TS is the temporal process modeling part of our model.DFTC-SF is the content feature modeling part of our model.
- **DFTC-SM**. DFTC-SM merge temporal process and content feature modeling via vector concatenation.

Metrics We adopt the accuracy and F1-score as metrics for the classification performance. Accuracy is the ratio of accurate predictions to all predictions. In our experiment, we study F1-scores of popularity levels by taking each of them as positive and other two levels as negative respectively.

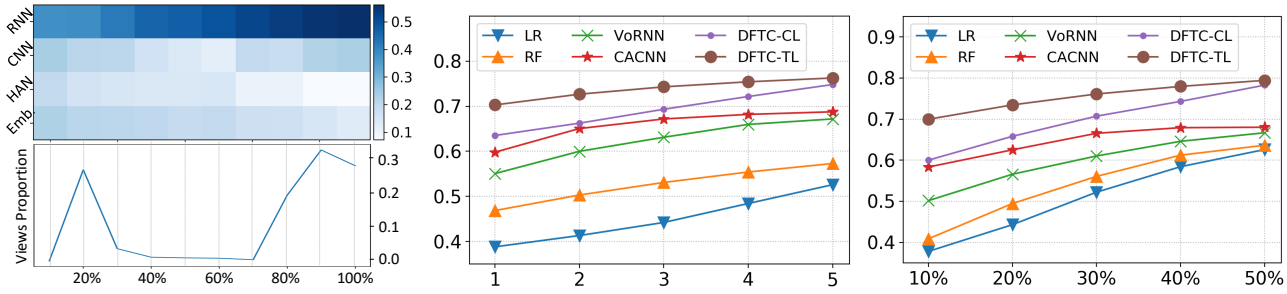
Parameter settings In temporal process modeling, we adopt a single LSTM layer with hidden size 512. We employ 4 kinds of kernels with sizes of 1,3,7,11 respectively. The number of each kind of kernels is set as 128. The CNN input window size k is 3 hours. In content feature modeling, we start with a pre-trained HAN for a relevant classification task and fine-tune parameters with our dataset. Embedding sizes of meta-data features are set as 32. We employ 2 FC layers for meta-data embedding combination, 1 FC layer for aligning $\boldsymbol{h}_t^r, \boldsymbol{h}_t^c, \boldsymbol{h}^h, \boldsymbol{h}^e$, and 1 FC layer after attention layer. All of FC layers employ ReLU as active function and have unified hidden size 512. In the decayed loss function, we set γ as 12. At last, we leverage the Xavier initialization and Adam optimizer for parameters learning, and employ dropout on each FC and RNN layers for regularization.

Complexity analysis As for our neural network, the major space consumption is the storage of weight matrix and the major time cost is the flops of linear transformation. Let n represent the max hidden size in the network, the space and time complexity of our model are both $O(n^2)$. In some other neural networks such as 2D-CNN for CV tasks, the intermediate results also consume lots of space. However, the intermediate results of our model is 1D-vector with max size n, which has no need for major consideration. Concretely, under our experiment setting, the total number of parameters is 6.2M. In the online prediction, the HAN and embedding part of our model compute only once for each specific article. The LSTM, attention CNN and attention fusion part of our model will be executed for each time step. For comparing time cost of the proposed model with LSTM and attention CNN, we conduct experiments based on TensorFlow with Tesla P40 GPU. One step forward prediction with batch size 64 consumes 0.29ms for CNN, 0.49ms for RNN and 0.97ms for our proposed DFTC.

Result Analysis

Comparison with Baselines Table 2 shows the prediction performance comparison measured by accuracy and F1-score. We can observe that the proposed method achieves the best result through all metrics on both test sets. Note that all methods get better results on the random test set. Under the realistic distribution, most articles' view amount are far away from the view amount of articles in other categories. Most "cold" articles have less than 10 views and most "normal" articles have less than 1000 views, as shown in Table 1. In balanced test set, there are much more examples closed to the classification boundary, which is more convincing for performance evaluation.

Specifically, feature-based classifiers' performances heavily depend on extracted features, which are hard to design and measure, especially for time series and text content. Thus they get the worst results. Hawks intensity processes only achieves the similar performance with feature-based classifiers, since the specific assumption of popularity dynamic limits the expressive power. Volume RNN models the popularity growth trend with LSTM, rather than a empirical function form. It outperforms feature-based classifiers and HIP, while it still lacks of predictive capability without regard to text and meta-data features. CACNN performs a little bit better than other state-of-the-art baselines in balanced test set, integrating both temporal process and meta-data feature modeling. However, the input of CNN is only a part of temporal process, which can not capture the long term growth trend of popularity. Comparing with them, the proposed DFTC



(a) Case study of $\boldsymbol{\alpha}^m$ (b) Accuracy of first 5 hours (c) Accuracy of first 50%

Figure 5: Performance Analysis. (a) The heatmap of average attentive weights $\boldsymbol{\alpha}^m$ and the line chart of short term fluctuations per 10% of time series. The weight of RNN increases over time and the weight of CNN corresponds to fluctuations. (b) Average accuracies of first 5 hours after articles are published. (c) Average accuracies when the observed popularity runs up to 10%-50% of the overall popularity.

model captures long term trend and short term fluctuation of the temporal process, and integrates text and meta-data features dynamically. Based on these modeling techniques, DFTC method significantly outperforms state-of-the-art methods. In balanced test set, it shows 17.0% increase of accuracy and 37.4%, 12.0%, 13.5% improvement of hot, normal and cold F1-score respectively. Our model have been applied in article recommendation in WeChat now.

Ablation Analysis The proposed model is also superior to its variants, i.e. DFTC-TS, DFTC-SF and DFTC-SM. DFTC-TS and DFTC-SF only leverage temporal processes or content features for prediction. It is no surprising that their performances are not desirable. Note that DFTC-TS outperforms other temporal process methods, i.e. VoRNN-TS and HIP, illustrating that the design of LSTM for long term trend and CNN for short term fluctuation is effective and superior. DFTC-SM combines temporal process and content feature modeling through vector concatenation, which lacks flexibility for handling the dynamic growth of time series. In contrast, DFTC method adopts attention mechanism to learn flexible weights for dynamic fusion. Thus it outperforms DFTC-SM by 7.8% on accuracy in balanced test set.

Case study In order to study effects of attention CNN and temporal attentive fusion, we randomly select a "hot" article as case study. We average its attentive weights $\boldsymbol{\alpha}^m$ and aggregate its view amount per 10% of time series, shown in Fig. 5a. The four rows of upper heat map represent attentive weights $\boldsymbol{\alpha}^m$ of RNN, CNN, HAN and embedding respectively. The bigger the weight is, the darker the color is. It can be observed that all these parts make similar contribution for prediction at the beginning. As the observed popularity getting closer to the overall popularity, RNN can learn the long term growth trend more accurately. Thus RNN model plays a major role in prediction at the later period. The lower line chart shows short term fluctuations of this article. We can observe the view amount shows outbreaks at the beginning and end of time series. Accordingly, attention CNN has high attentive weights at the beginning and the end, which means it effectively captures such local rising and falling structures that are important to the overall popularity prediction.

Performance in Early Stage In practical applications, it is much valuable to predict overall popularity in early stage. Figure 5b shows average accuracies of first 5 hours after articles are published and Fig. 5c shows average accuracies when the observed popularity runs up to 10%-50% of the overall popularity in balanced test set. We can observe that our model shows an obvious improvement of the early stage prediction performance, from both time and observed popularity aspects. Note that the performance of HIP is not shown here, because it needs a long enough temporal process to estimate parameters well (Rizoiu et al. 2017). Feature-based methods perform very poorly in early stage, since handcrafted features can hardly capture temporal dynamic patterns. CACNN integrates meta-data features for prediction, thus it performs better than VoRNN-TS in early stage.

DFTC-CL is our model with the common sequence loss. Content feature modeling enables our method to make reliable predictions when the temporal process lacks enough information, and the attentive fusion ensures content feature modeling plays a major role in early stage. Thus DFTC-CL significantly outperforms state-of-the-art methods. The temporal decayed factor of loss function helps our model to invest more efforts to optimize the prediction performance of early stages. When we apply the temporal decayed loss function for optimizing, i.e. DFTC-TL, the early stage performance is further improved. Besides, since the observed popularity gets closer to overall popularity over time, it can also get a desirable performance in the latter period, even if we reduce their weights.

Conclusion

In this work, we propose a novel online articles popularity prediction method. We adopt RNN for capture the long term trend and CNN to extract short term fluctuation. We exploit HAN to model text features and employ embedding techniques to learn meta-data features. At last, a temporal attention fusion layer is employed to integrate all these parts dynamically. Evaluation results based on real-world online article dataset demonstrate the effectiveness and superiority of the proposed model.

Acknowledgment

This work was done when Dongliang Liao was interning at WeChat, Tencent Inc. We would like to thank our WeChat colleague Shen Huang for his great help in experiments and writing of this paper. We would also like to thank our WeChat colleagues Zhe Feng, Yuetang Deng, Zhiping Wang and Yandong Bai for useful discussions and supports of this work. We would also thank Prof. Feiping Nie of Northwestern Polytechnical University and Prof. Yudong Chen of Cornell University for their valuable suggestion.

References

Cao, Q.; Shen, H.; Cen, K.; Ouyang, W.; and Cheng, X. 2017. Deephawkes: Bridging the gap between prediction and understanding of information cascades. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1149–1158. ACM.

Du, N.; Dai, H.; Trivedi, R.; Upadhyay, U.; Gomez-Rodriguez, M.; and Song, L. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1555–1564. ACM.

Gao, H.; Kong, D.; Lu, M.; Bai, X.; and Yang, J. 2018. Attention convolutional neural network for advertiser-level click-through rate forecasting. 1855–1864.

Gou, C.; Shen, H.; Du, P.; Wu, D.; Liu, Y.; and Cheng, X. 2018. Learning sequential features for cascade outbreak prediction. *Knowledge and Information Systems* 1–19.

Huo, X.; Li, M.; and Zhou, Z.-H. 2016. Learning unified features from natural and programming languages for locating buggy source code. In *IJCAI*, 1606–1612.

Keneshloo, Y.; Wang, S.; Han, E.-H.; and Ramakrishnan, N. 2016. Predicting the popularity of news articles. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, 441–449. SIAM.

Lai, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, 2267–2273.

Liu, A.-A.; Nie, W.-Z.; Gao, Y.; and Su, Y.-T. 2016. Multi-modal clique-graph matching for view-based 3d model retrieval. *IEEE Transactions on Image Processing* 25(5):2103–2116.

Mishra, S.; Rizoiu, M.-A.; and Xie, L. 2018. Modeling popularity in asynchronous social media streams with recurrent neural networks. *arXiv preprint arXiv:1804.02101*.

Piotrkowicz, A.; Dimitrova, V.; Otterbacher, J.; and Markert, K. 2017. Headlines matter: Using headlines to predict the popularity of news articles on twitter and facebook. In *ICWSM*, 656–659.

Rizoiu, M.-A.; Xie, L.; Sanner, S.; Cebrian, M.; Yu, H.; and Van Hentenryck, P. 2017. Expecting to be hip: Hawkes intensity processes for social media popularity. In *Proceedings of the 26th International Conference on World Wide Web*, 735–744. International World Wide Web Conferences Steering Committee.

Sanjo, S., and Katsurai, M. 2017. Recipe popularity prediction with deep visual-semantic fusion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2279–2282. ACM.

Shen, H.-W.; Wang, D.; Song, C.; and Barabási, A.-L. 2014. Modeling and predicting popularity dynamics via reinforced poisson processes. In *AAAI*, volume 14, 291–297.

Shulman, B.; Sharma, A.; and Cosley, D. 2016. Predictability of popularity: Gaps between prediction and understanding. In *ICWSM*, 348–357.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Xu, J.; Yin, Y.; Man, H.; and He, H. 2012. Feature selection based on sparse imputation. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, 1–7. IEEE.

Xu, J.; Tang, B.; He, H.; and Man, H. 2017. Semisupervised feature selection based on relevance and redundancy criteria. *IEEE transactions on neural networks and learning systems* 28(9):1974–1984.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1480–1489.

Yu, H.; Xie, L.; Sanner, S.; et al. 2015. The lifecyle of a youtube video: Phases, content and popularity. In *ICWSM*, 533–542.

Zhang, W.; Wang, W.; Wang, J.; and Zha, H. 2018. User-guided hierarchical attention network for multi-modal social image popularity prediction. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, 1277–1286. International World Wide Web Conferences Steering Committee.

Zhao, Q.; Erdogdu, M. A.; He, H. Y.; Rajaraman, A.; and Leskovec, J. 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1513–1522. ACM.