A named entity topic model for news popularity prediction

Knowledge-Based Systems 208 (2020) 106430

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A named entity topic model for news popularity prediction

Yang Yang [a], Yang Liu [c], Xiaoling Lu [b,c], Jin Xu [a], Feifei Wang [b,c,*]

[a] School of Electronic Engineering and Computer Science, Peking University, China
[b] Center for Applied Statistics, Renmin University of China, China
[c] School of Statistics, Renmin University of China, China

ARTICLE INFO

Article history:
Received 14 May 2020
Received in revised form 1 August 2020
Accepted 10 September 2020
Available online 17 September 2020

Keywords:
Named entity
Web content
Popularity prediction
Topic model

ABSTRACT

Predicting the popularity of web content is widely regarded as an important but challenging task. Online news articles are typical examples of this. In particular, owing to their time-sensitive nature, it is preferable to predict the popularity of news articles before their publication. To achieve this, this study proposes a named entity topic model (NETM) to extract the textual factors that can drive popularity growth. Here, each named entity is assumed to have a popularity-gain distribution over all semantic topics. The popularity of a news article is considered as the accumulation of popularity gains generated by its named entities (NEs) over all the topics. By learning the popularity-gain matrix for each named entity, the popularity of any news article can be predicted. Experiments on two collections of news articles demonstrate that the proposed NETM can outperform existing models in terms of accuracy. Additionally, the popularity-gain matrix learned by the NETM can be used to effectively explain the popularity of specific news articles.

1. Introduction

The popularity of web content is of notable importance in commercial contexts [1–6]. For example, forecasting and monitoring the view count of streaming videos can help to improve targeted advertising strategies [7]. Similarly, investigating tweet propagation can help to facilitate political campaigns by improving the quality of promotional copywriting and alternative propaganda strategies [4]. Therefore, web content popularity prediction has received significant attention in both research and industrial fields.

Web content is characterized by its rapid rate of propagation. Therefore, its popularity must be predicted within a short period after the content is released to the public. To this end, early popularity growth trends, such as those in view counts and propagation on social networks, are taken as strong indicators for future popularity gain [2,6,8,9]. Online news articles are some of the most time-sensitive forms of web content. Therefore, it is preferable to forecast the popularity of these articles before publication. In such a situation, the dynamics of early popularity growths are unavailable, putting the prediction task in a “cold start" state.

Predicting the popularity of web content in a cold-start condition is a challenging task [10,11]. When early popularity growth data are unavailable, it is essential to explore content features that can drive or indicate popularity growth. For example, Bandari et al. extracted various content-based features (including language subjectivity, named entity (NE) prominence, and source authority) to classify news articles into different preset popularity classes [2]. Arapakis et al. found that articles with more than one NE are more likely to become popular than those with no NEs [10]. They extracted NEs from news articles, tracked their popularities on Twitter, Wikipedia, and various web search engines, and then used them as indicators for popularity prediction [10].

Over the past few decades, topic models have achieved great success in obtaining the semantic meanings underlying the text documents. Therefore, the use of topic models for web content popularity prediction has received increasing attention in recent years [12–14]. One notable work is that by Dou et al. who proposed the linkage of online items with existing knowledge-based entities and leveraged embedded entity relationships as indicators to improve popularity prediction [15]. Additionally, Abbar et al. utilized topic popularity prediction to improve news article popularity prediction [16].

Although research has found that NEs and semantic topics influence the popularity of web content, the interactions between these two factors are yet to be thoroughly explored. However, such interactions can often be observed. Consider the following three examples of headlines of news articles published in the NetEase News, which is one of the most popular Chinese news publishing platforms:

* Correspondence to: Center for Applied Statistics, School of Statistics, Renmin University of China, China.
E-mail address: feifei.wang@ruc.edu.cn (F. Wang).

https://doi.org/10.1016/j.knosys.2020.106430
0950-7051/© 2020 Elsevier B.V. All rights reserved.

- *The food is great! Miranda Kerr was going out with a pregnant belly and rounder body.* – 356 views.[1]
- *Taylor Swift is getting fat again? Pink dancing shorts with hot thighs thicker than a circle* – 613 views.[2]
- *Taylor Swift donates to fans facing financial difficulties* – 84 views.[3]

The first two news articles both discussed the topic of weight gain, but related to different celebrities, i.e., the supermodel Miranda Kerr and the famous singer Taylor Swift, and thus triggered significantly different views. The second and third news articles were both related to Taylor Swift but discussed different topics, each drawing different levels of attention. From these two comparisons, it can be seen that a topic associated with different NEs, or the association of an NE with different topics, may influence web popularity differently. Therefore, it can be inferred that exploring the relationship between NEs and semantic topics would provide a novel perspective of factors affecting popularity gain.

Under this assumption, this study proposes a novel NE topic model (NETM) to learn popularity-related factors from news articles. Specifically, for each NE, there exists a popularity-gain distribution over all the topics. For each topic, there exists a word distribution over the entire vocabulary in the corpus. The popularity of a specific article is assumed to be positively correlated with the accumulation of popularity gain of its NEs over its topics. Essentially, when the interaction of named entities and topics in a news article has a higher popularity gain, the popularity of the article is inclined to increase. Therefore, the key principle of NETM is to learn the "NE topic" popularity-gain matrix. Once this matrix is obtained, the popularity of any given news article, including those in conceptual stages, can be forecasted.

The remainder of this paper is organized as follows. Section 2 reviews existing works related to popularity prediction. Section 3 introduces the proposed NETM and its inferences. Section 4 provides two real news article datasets. The experimental settings in this study are also presented. Section 5 illustrates the NE topic popularity-gain matrix. Section 6 presents the experimental results in detail. Section 7 discusses the model efficiency. Finally, the relevant conclusions are discussed in Section 8.

2. Related work

2.1. Pre-publication popularity prediction

Pre-publication popularity prediction for news articles has already received significant attention from researchers. Early works by Tsagkias et al. [8] considered this problem to be a two-step classification task. They first classified articles based on the presence or absence of comments and then defined the specific popularity values according to the number of received comments. For popularity prediction, they extracted a set of surface, cumulative, textual, semantic, and real-world features from the textual content. These proved to be strong performers. Bandari et al. [2] further considered four characteristics of the articles: news source, news category, language subjectivity, and NEs. They reported that popularity on social media could be predicted with 84% accuracy using bagging techniques.

With the exception of textual features extracted from news articles, Arapakis et al. [10] utilized more features from external sources as popularity indicators. For example, they considered the popularity of NEs on Twitter, Wikipedia, and web search engines as external features. Fernandes et al. [17] proposed a proactive intelligent decision support system (IDSS) to achieve pre-publication prediction. Their work extended previous studies by implementing groups of linguistic features while simplifying the popularity prediction task as a binary classification problem. This system achieved 73% popularity prediction accuracy on a Mashable news dataset using a random forest algorithm with a rolling-window strategy. Uddin et al. [11] regarded user shares as a news popularity index. Based on a public dataset of news articles, they found that features extracted from article keywords, publication dates, and the data channel significantly influence popularity prediction. Further, they achieved a 1.8% improvement over the IDSS proposed by Fernandes et al. [17].

2.2. Popularity prediction with NEs and semantic topics

Features extracted from NEs and topics in news articles are broadly used as powerful indicators for popularity prediction. Gelli et al. [18] used visual sentiment features together with contextual features to predict the popularity scores of social images. The contextual features in their work included domain features and type features of top-ranked topics in the free base, obtained by corresponding image tags, and occurrences of seven-class NEs extracted from image descriptions. The proposed method showed that contextual features together with sentiment features and user features could achieve good prediction performance. Keneshloo et al. [5] considered the prediction problem in terms of linear regression. They extracted metadata, contextual, temporal, and social features, and built models to forecast the page view count of news in *The Washington Post*.

Among the contextual features, the number of NEs in news articles was used as an indicator to explore the extent to which an article discussed a subject. This proved to be helpful to the prediction performance. Piotrkowicz et al. [19] engineered two types of features from news headlines: (1) journalism-inspired news values, including entity prominence, sentiment, magnitude, proximity, surprise, and uniqueness, and (2) linguistic style, including brevity, simplicity, unambiguity, punctuation, nouns, verbs and adverbs. Based on these features, they applied support vector regression (SVR) with the radial basis function kernel to predict the popularity of news articles on Twitter and Facebook. The proposed method significantly outperformed several baselines, and the corresponding features were shown to impact the prediction performance. Dou et al. [15] linked online entities with existing knowledge-based entities, and proposed a novel prediction model based on long short-term memory (LSTM) networks. By adaptively incorporating knowledge-based embedding of the target entity, as well as the popularity dynamics from items with similar entity information, the LSTM achieved good performance in terms of web content popularity prediction. Abbar et al. [16] found that the popularity of a topic depends on that of related topics, while the popularity of an article depends on that of similar, recently published articles. Based on these findings, they extended their approach and used topic popularity prediction to improve news article popularity prediction.

2.3. Joint modeling of topics and other popularity factors

Authorships, co-authorships, textual sentiment, and web content popularity are inherently related to each other. Thus, they can be modeled together. For example, Liu et al. [20] developed a Bayesian hierarchical approach to perform topic modeling and author community discovery in one unified framework. They found a set of high-level topics covered by the documents in the collection. Li et al. [21] proposed the tag resource latent Dirichlet allocation (TTR-LDA) community model, which combines the

[1] http://ent.163.com/photoview/00AJ0003/650914.html#p=DDT0A9LT00AJ0003NOS.

[2] https://dy.163.com/article/EN9IP5QM05444NSH.html.

[3] https://3g.163.com/3g/article_cambrian/F8L999SM05129QAF.html.

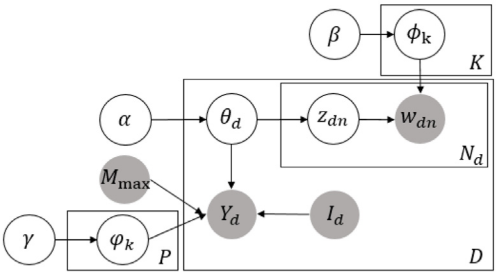


Fig. 1. Generative process of NETM. Y_d, I_d, w_{dn} are observed data, and $M_{\max}$ is a pre-defined value.

TTR-LDA model [22] and the Girvan–Newman community detection algorithm [23]. TTR-LDA is a three-layer Bayesian model, including the taggers layer, resource layer, and latent topics layer. It is effective for understanding communities in tag prediction. Han et al. [24] proposed an entity-topic generative model for entity linking. By uniformly modeling context compatibility, topic coherence, and their correlations, the model can accurately link all mentions in a document using both local information and global knowledge. Lin et al. [25] proposed a sentiment-topic model, a supervised joint model of sentiments and topics. This model assumes there exists a document-specific sentiment distribution. It is useful for classifying documents and obtaining more coherent and informative topics.

3. NETM

This study proposes an NETM to learn the factors affecting the popularity of published news articles. By jointly modeling NEs and topics extracted from the article content, the NETM can predict pre-publication popularity. Specifically, the model assumes that the popularity of a given article is positively correlated with the accumulation of popularities achieved by every NE it contains. For each NE, there exists a popularity-gain distribution over all the topics, and the popularity gained is derived from the inner product of the popularity-gain distribution and topic distribution. Intuitively, it can be understood that when NEs achieve higher popularity gain over major topics of a news article, the article tends to gain higher popularity.

3.1. Model description

The NETM builds upon the principles of topic modeling techniques; it also introduces NEs to more effectively capture the popularity of general articles. Assume there are K topics underlying D documents. Each document d has a distribution $\boldsymbol{\theta}_d = \{\theta_{d1}, \theta_{d2}, \dots, \theta_{dK}\}$ over K topics. Each topic k has a distribution $\boldsymbol{\phi}_k = \{\phi_{k1}, \phi_{k2}, \dots, \phi_{kV}\}$ over V words, which is the number of all distinct words appearing in the corpus. Assume there are P NEs in the entire corpus, and each NE p has an “NE topic" distribution $\boldsymbol{\varphi}_p = \{\varphi_{p1}, \varphi_{p2}, \dots, \varphi_{pk}\}$ over K topics. For each document d, a P-dimensional binary vector $\boldsymbol{I}_d = \{I_{d1}, I_{d2}, \dots, I_{dP}\}$ is used to indicate the presence of the pth NE. Meanwhile, a popularity variable Y_d is used to represent the accumulated topic popularity for d. Here, Y_d is assumed to be a count variable. For example, in the NetEase and Tencent datasets used in Section 4, the popularity indicators are the number of views and the number of viewer comments.

Based on the above assumptions, Fig. 1 presents the generative process of the popularity of the document d.

1. For each topic k, generate word probabilities $\boldsymbol{\phi}_k \sim \text{Dir}(\beta)$ over a dictionary space of size V.
2. For each NE p, generate NE topic probabilities $\boldsymbol{\varphi}_p \sim \text{Dir}(\gamma)$ over all the K topics.
3. For each document d, $d \in \{1, 2, \dots, D\}$:
   (a) Generate topic distribution $\boldsymbol{\theta}_d$ from a homogeneous Dirichlet distribution with hyper-parameter α, i.e., $\boldsymbol{\theta}_d \sim \text{Dir}(\alpha)$.
   (b) For the nth word in the document, $n \in \{1, 2, \dots, N_d\}$
      i. Choose a topic z_{dn} with probabilities given by $\boldsymbol{\theta}_d$, i.e., $z_{dn} \sim \text{Multi}(\boldsymbol{\theta}_d)$.
      ii. From the dictionary, choose a word w_{dn} with probabilities given by $\boldsymbol{\phi}_{z_{dn}}$, i.e., $w_{dn} \sim \text{Multi}(\boldsymbol{\phi}_{z_{dn}})$.
   (c) The popularity gain λ_d is given by
   $$\lambda_d = \sum_{k=1}^{K}\sum_{p=1}^{P}\theta_{dk}I_{dp}\varphi_{pk}.$$
   (d) Finally, the observed popularity of document d is generated from the Poisson distribution:
   $$Y_d = \text{Poisson}(\lambda_d \times M_{\max}),$$
   where $M_{\max}$ is the upper limit popularity value, which must be specified in advance. In practice, $M_{\max}$ can be gained and updated from the data.

3.2. Model inference

This section introduces the Gibbs sampling algorithm for NETM. Let $\boldsymbol{Y} = (Y_1, Y_2, \dots, Y_D)^\top$, $\boldsymbol{z}_d = (z_{d1}, z_{d2}, \dots, z_{dN_d})^\top$, $\boldsymbol{z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_D\}$, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_D\}$, $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \dots, \boldsymbol{\phi}_K\}$, $\boldsymbol{\Psi} = \{\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \dots, \boldsymbol{\varphi}_K\}$ and $\boldsymbol{I} = \{\boldsymbol{I}_1, \boldsymbol{I}_2, \dots, \boldsymbol{I}_D\}$. Let $\boldsymbol{w}$ be the collection of all words. Then, given the hyper-parameters (α, β, γ) and observed data $(\boldsymbol{Y}, \boldsymbol{w}, \boldsymbol{I})$, the full posterior distribution of NETM can be derived as follows:

$$\begin{aligned} & f(\boldsymbol{z}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{\Psi} | \boldsymbol{Y}, \boldsymbol{w}, \boldsymbol{I}, \alpha, \beta, \gamma) \\ \propto & \left\{\prod_{d=1}^{D}\prod_{k=1}^{K}\theta_{dk}^{\alpha-1}\right\}\left\{\prod_{k=1}^{K}\prod_{v=1}^{V}\phi_{kv}^{\beta-1}\right\}\left\{\prod_{p=1}^{P}\prod_{k=1}^{K}\varphi_{pk}^{\gamma-1}\right\} \\ \times & \left\{\prod_{d=1}^{D} f(Y_d|\lambda_d) f(\boldsymbol{\theta}_d|\alpha)\prod_{n=1}^{N_d} f(z_{dn}|\boldsymbol{\theta}_d) f(w_{dn}|z_{dn}, \boldsymbol{\Phi})\right\} \\ \times & \left\{\prod_{k=1}^{K} f(\boldsymbol{\phi}_k|\beta)\right\}\left\{\prod_{p=1}^{P} f(\boldsymbol{\varphi}_p|\gamma)\right\} \end{aligned} \tag{1}$$

where $f(Y_d = y|\lambda_d) = \exp(-M_{\max}\lambda_d)\frac{(M_{\max}\lambda_d)^y}{y!}$, and $\lambda_d = \sum_{k=1}^{K}\sum_{p=1}^{P}\theta_{dk}I_{dp}\varphi_{pk}$.

Given the full posterior distribution in Eq. (1), the full conditional posterior distributions for $\boldsymbol{z}$, $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\Psi}$ are easily obtained. For the nth word in the document d, the full conditional posterior distribution of z_{dn} is given by

$$f(z_{dn} = k|\cdot) \propto \theta_{dk}\phi_{k,w_{dn}} \tag{2}$$

For $\boldsymbol{\phi}_k$, $k \in \{1, 2, \dots, K\}$, the full conditional posterior distribution is

$$\begin{aligned} f(\boldsymbol{\phi}_k|\cdot) & \propto \left\{\prod_{v=1}^{V}\phi_{kv}^{\beta-1}\right\}\left\{\prod_{d=1}^{D}\prod_{n=1}^{N_d}\phi_{z_{dn},w_{dn}}\right\} \\ & \propto \prod_{v=1}^{V}\phi_{kv}^{\beta-1+n_{kv}^{(1)}}, \end{aligned} \tag{3}$$

where $n_{kv}^{(1)}$ is the number of times a word v is associated with a topic k in corpus.

For $\boldsymbol{\varphi}_p, p \in \{1, 2, \ldots, P\}$, the full conditional posterior distribution is

$$f(\boldsymbol{\varphi}_p|\cdot) \propto \left\{\prod_{k=1}^{K} \varphi_{kp}^{\gamma-1}\right\} \left\{\prod_{d=1}^{D} f(Y_d = y|\lambda_d)\right\} \tag{4}$$

For $\boldsymbol{\theta}_d, d \in \{1, 2, \ldots, D\}$, the full conditional posterior distribution is

$$\begin{aligned} f(\boldsymbol{\theta}_d|\cdot) \propto & \left\{\prod_{k=1}^{K} (\theta_{dk})^{\alpha-1}\right\} \left\{f(Y_d = y|\lambda_d) \prod_{n=1}^{N_d} \theta_{d,z_{dn}}\right\} \\ & \propto f(Y_d = y|\lambda_d) \left\{\prod_{k=1}^{K} (\theta_{dk})^{\alpha-1+n_{dk}^{(2)}}\right\}, \end{aligned} \tag{5}$$

where $n_{dk}^{(2)}$ is the number of words associated with the topic k in the document d.

The full conditional posterior distributions of z_{dn} and $\boldsymbol{\phi}_k$ are multinomial and Dirichlet, respectively, and thus, they can be sampled directly. However, it is difficult to obtain samples from the full conditional posterior distributions of $\boldsymbol{\varphi}_p$ and $\boldsymbol{\theta}_d$. Here, Metropolis–Hastings algorithms are used to handle this issue. Specifically, a univariate version of this algorithm is used initially; the proposal distribution for each target variable is a univariate normal distribution, which is centered at the current value and has some variance σ^2. Then, the variance σ^2 is tuned to produce a Metropolis acceptance rate between 15 and 40%.

4. Data and experimental setup

4.1. Datasets

The effectiveness of NETM was evaluated over two real news datasets. The first dataset was collected from NetEase News (http://news.163.com/), which is one of the most popular Chinese news publishing platforms (hereinafter referred to as NetEase). It contains 357,921 news articles, published between June 1, 2016 and September 30, 2018. The second dataset was collected from Tencent News (http://news.qq.com/) (hereinafter referred to as Tencent). It includes 87,083 news articles, published between June 1, 2018 and March 1, 2019. To illustrate the popularity of the news articles, the NetEase dataset includes both the number of views and the number of viewer comments. In the Tencent dataset, only the number of viewer comments is available. For both datasets, the number of views and viewer comments were collected seven days after the publication of a news article, by which time the popularity growths of most news articles have stagnated.

Fig. 2 illustrates the distributions of the number of views and viewer comments in both datasets. Specifically, the top three subfigures present histograms of original values on the logarithm scale. It is clear that all three histograms are skewed, indicating the existence of extremely popular news articles. In addition, the number of views in NetEase has the widest data range, whereas that in Tencent has the narrowest. In both datasets, there is also a large proportion of news articles with zero popularity (around 40%), which we refer to as *silent* news articles. To clearly illustrate the distribution of non-zero popularities, the bottom three sub-figures show histograms of different popularity indicators omitting zeros. As shown, the NetEase dataset has more silent news articles than the Tencent one. In the experiments conducted in this study, the prediction performance of NETM was evaluated for datasets with or without zero popularity.

First, the common text-mining practice of removing digits, punctuation marks, and English words from the news articles was followed. Then, because both datasets are in Chinese, word segmentation had to be performed to obtain the contexts of the news articles. This was done using an open-source package *Jieba*. Following the word segmentation, stopwords and low-frequency words were removed. To extract the NEs from the textual matter, the *HIT-SCIR/LTP toolkits* [26] were applied. These toolkits can help to obtain three types of NEs: people, places, and organizations. Following the data preprocessing, the NetEase dataset had 1,043,094 unique Chinese words and 522,137 NEs, while the Tencent dataset had 386,014 unique Chinese words and 119,071 NEs. A statistical overview of the NetEase and Tencent datasets are summarized in Table 1.

Table 1
Basic statistics of NetEase and Tencent datasets.

Dataset	NetEase	Tencent
# of docs	357,921	87,083
# of words	1,043,094	386,014
Avg words per doc	327.39	288.81
# of NEs	522,137	119,071
Avg NEs per doc	18.32	13.38

4.2. Experimental setup

Before applying NETM on the news datasets, cluster analysis was conducted on the collection of NEs. Compared with the number of topics, which is usually at the scale of tens or hundreds, the number of NEs is often so large that learning the NE topic probability matrix would result in a severe sparsity problem. To address this issue, the millions of NEs were compressed into thousands of NE clusters, and these clusters were used in place of the NEs in the NETM. Specifically, the semantic meanings of the NEs were first mapped to word vectors using the Tencent AI Lab Embedding Corpus [27], which provides 200-dimensional vector representations for over eight million Chinese words and phrases. Then, the embedded word vectors were clustered into groups using the K-means algorithm. The number of NE clusters (n_cluster) was selected using the gap statistics. For example, in the overall popularity prediction, the resulting optimal numbers of clusters for NetEase and Tencent were 2250 and 1650, respectively. It may be noted that, when using NE clusters rather than NEs in the NETM, the number of parameters in the NETM can be significantly reduced. Furthermore, the problem of learning popularity gain for new emerging NEs can be effectively addressed, as they can be classified into the existing NE clusters.

Following the cluster analysis, NETM was applied to both NetEase and Tencent. The model performance was compared with four baseline methods and three state-of-the-art methods. To ensure a reasonable comparison, the data utilized by all competitors were limited to news article titles and content. Specifically, the four baseline methods are as follows:

- **Support vector machine with NEs (SVM-NE).** Only the NE clusters are used as features in an SVM for popularity prediction.
- **Supervised LDA (S-LDA).** In supervised LDA [28], the popularity of news articles is taken as the dependent variable, while the extracted topics are taken as independent variables.
- **LSTM.** The LSTM model is proposed to explore the relationship between web content and popularity. In this model, an LSTM layer extracts features from embedded news content; it is then combined with dense layers to enable prediction.

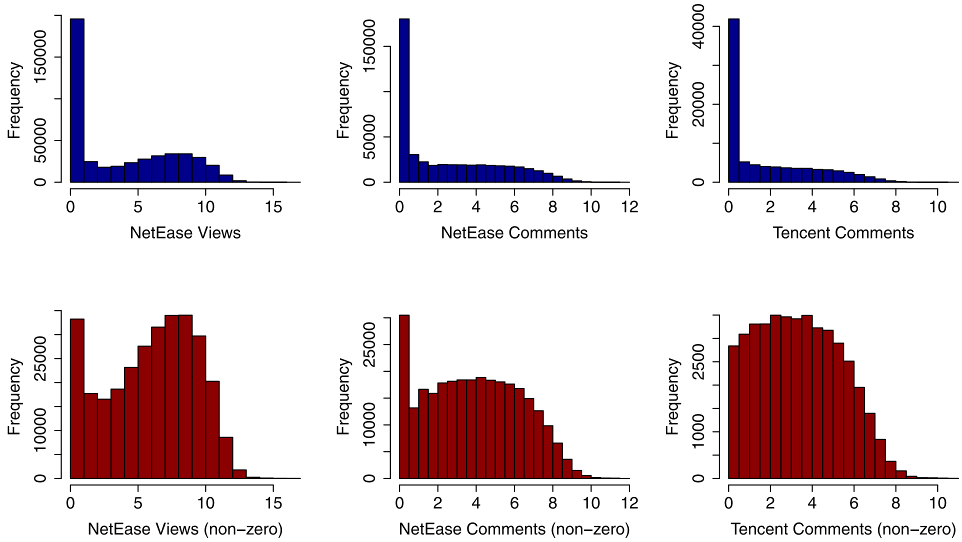


Fig. 2. Histograms representing the logarithm of the number of views in NetEase, number of viewer comments in NetEase, and number of viewer comments in Tencent, respectively. The top three figures (marked in blue) describe the original values, while the bottom three figures (marked in red) describe the values after omitting zero. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- **Feature Collection (FC).** From previous works, such as [5,8,10], a range of timing, cumulative, semantic, readability, sentiment, and textual features are extracted. Thus, 39 suitable features are obtained for reasonable comparison. A random forest predictor is then applied to make predictions using these features.

The state-of-the-art methods include the following:

- **Integration of Topic Popularity and Article Popularity (TPAP) Forecasting.** For an article, the TPAP approach exploits the popularity of its related topics and recent similar articles to achieve early prediction of its popularity [16]. To fully utilize its predictive power, its full feature groups, including article similarity, topic volume, and predicted topic volume, were chosen for prediction. It may be noted that TPAP exploits the time series of popularity for each article, whereas the popularity indicators in the NetEase and Tencent datasets were collected when the popularity growth was stagnant. However, because their time serial features are extracted with topic granularity, the TPAP approaches can still be applied to these two datasets.
- **Headlines Matter (HM).** The HM method utilizes a wide variety of text features extracted from news headlines to predict the popularities of news articles [19]. The majority of textual features need to be extracted using special natural language processing tools, which are primarily designed for English; therefore, before the downstream feature extraction process, the Chinese news titles were first translated into English through the Google Translate service. However, it should be noted that following the translation, the HM method would undergo significant information loss and statistical shifting in the distributions of lexical, syntactical, semantical, and sentimental feature groups. Additionally, as most NetEase and Tencent readers originate from Mainland China, the original geographic proximity feature in HM is omitted. As a result, 31 out of the 32 headline features were recovered from HM for popularity prediction.
- **Intelligent Decision Support System (IDSS) for Popularity Prediction.** The IDSS method considers a binary popularity prediction task [17]. It uses a broad set of extracted features (including keywords, digital media content, and prior popularity of news referenced in the article) to predict whether a specific article will become popular in the future. Because there are no keywords and internal reference links in the NetEase and Tencent datasets, 32 out of the 47 original features can finally be reproduced. Moreover, to make the IDSS method suitable for the prediction task in this study, SVR was used instead of SVM for popularity prediction.

For all the methods used for comparison, the original experimental settings were largely followed. For NETM, all hyperparameters (e.g., α, β) were selected via a grid search. For both the NetEase and Tencent datasets, the news corpus is divided into two parts, i.e., the training dataset (80%) and the test dataset (20%). The models were built on the training dataset, following which their population prediction accuracies were evaluated on the test dataset. Specifically, let Y_d define the true popularity value of article d, and $\hat{Y}_d$ define the corresponding predicted popularity value. Then, two measures, the root-mean-square error (RMSE) and the mean absolute percent error (MAPE) were used to assess the popularity prediction performance, i.e.,

$$\begin{aligned} \mathrm{RMSE} &= \sqrt{\frac{1}{D}\sum_{d=1}^{D}(Y_d - \hat{Y}_d)^2}, \\ \mathrm{MAPE} &= \frac{1}{D}\sum_{d=1}^{D}|Y_d - \hat{Y}_d|/Y_d. \end{aligned} \tag{6}$$

It may be noted that, when calculating the MAPE for a news article with a popularity value of zero, the MAPE denominator Y_d changes to $Y_d + 1$. In practice, it is more meaningful to conduct accurate popularity prediction for well-received news articles, because they generally draw more public attention and have greater social impacts. Therefore, this work focuses on accurate predictions for highly popular news articles. Specifically, articles from the highest q popularity quantile were chosen, and the percentage of correctly predicted articles in the same quantile was calculated by the given prediction method.

5. NE clusters and NE topic probability matrix

To illustrate the performance of the NE clusters and NE topic probability matrix, the experiment on the NetEase dataset may be considered. By using gap statistics, the optimal number of NE

Table 2
Examples of five NE clusters in the NetEase dataset.

NE Cluster	Example NEs
National Leaders	Bernanke, Clinton, Cameron, Yeltsin, Kissinger, Junker, Nixon, Peter the Great, De Gaulle
Entrepreneurs	Richard Liu, Pony Ma, Jack Ma, Ren Zhengfei, Hui Ka Yan, Wang Jian lin, Chuanzhi Liu
European Football Clubs	Leverkusen, Cadiz, Lyon, Qarabag, Lugansk, Toulouse, Thessaloniki, Parma, Grozny, Murcia
Internet Celebrities	Miss, Gula Dai, Timo Feng, Papi, Jiaqi Li, Lige, Yifa Chen, Ziqi Li, Weiya, Xukun Cai
Cities	Rome, Beijing, Seoul, New York, Shenzhen, Paris, Xiamen, Paris, Canberra, San Jose

Table 3
Examples of topics in the NetEase dataset.

Topic	Top-ranked words
Illegal	illegal, punishment, drug abuse, legality, report, investigation, law, violation, order, marijuana
Sovereignty Dispute	Russia, Ukraine, army, EU, president, sanction, Moscow, border, refugee, negotiation, cease fire
Charity	relieve, donation, fund, trust, goodwill, entrepreneur, education, public, hope, NGO
Finance	marker, securities, fund, stock market, investor, stock, billion, index, broker, gain
Crisis	reconcil, bankruptcy, crime, hypocritical, eavesdrop, drug, corrupt, patent, victim, finance
Debate	bet, desire, focus, disagree, satisfied, insist, accusation, tolerant, anger, query
Football Competitions	game, sports, football, Olympic games, Brazil, athlete, club, champion, player, world cup
Cosmetic Surgery	cosmetic surgery, Korean, agency, models, hospital, nose, factory, performance, income, student
Scandal	gossip, drunk, track, inside story, cheat, love affair, drug abuse, crime, accuse, implicate
Public Relations	interest, team, money, business, economics, internet, desire, network traffic, ad, podcasting

clusters is 2250. In this experiment, the view count was taken as the popularity indicator, and we set $M_{\max} = 10^8$ and the number of topics as $K = 160$. The hyper-parameters (α, β, γ) were selected using a grid search.

5.1. NE clustering

To eliminate the sparsity problem in parameter learning, the large amount of individual NEs can be compressed into smaller NE clusters. Therefore, a well-performed NE clustering process contributes significantly to the subsequent model inference. To illustrate the NE clustering performance in this work, Table 2 lists five randomly selected NE clusters and some example NEs. Each NE cluster is named based on the NEs it contains. For example, the first cluster includes “Bernanke" (the former chairman of the Federal Reserve Committee), “Cameron" (the former British Prime Minister), “De Gaulle" (the former president, general of France), and “Peter the Great" (Czar of the Romanov Dynasty in Russia). As these NEs are all national leaders of different countries or different eras, this cluster was named *National Leaders*. Another representative cluster is *European Football Clubs*, which includes the “Leverkusen" club in Germany, the “Cadiz" club in Spain, the “Lyon" club in France, and the “Qarabag" club in Azerbaijan. Similar to these two clusters, as shown in Table 2, all NEs in a specific cluster are closely correlated, which suggests an explicit cluster meaning.

5.2. NE-topic probability matrix

Learning the NE topic probability matrix (i.e., $\boldsymbol{\varphi}_p$) is the principal assumption in popularity prediction. Here, the performance of NETM in achieving this is shown. Fig. 3 shows the popularity distributions over all the 160 topics for each of the five NE clusters present in Table 2. It is clear that varied probabilities exist over different topics for different NE clusters. For example, the NE cluster *National Leaders* shows particularly high popularity probabilities over two specific topics, marked in blue. As described in Table 3, these two topics are titled *illegal* and *sovereignty dispute* based on words with high probabilities (i.e., $\boldsymbol{\phi}_k$). Within the NE cluster *Entrepreneurs*, four topics (marked in red) have relatively high popularity probabilities. These are *charity*, *finance*, *scandal* and *debate*. The NE clusters *European Football Clubs* and *Internet Celebrities* also include certain topics with high probabilities. Specifically, the topics *finance*, *illegal*, and *football competition* (in yellow) are strongly related to the *European Football Clubs* cluster, while *cosmetic surgery*, *scandal*, and *public relations* (in purple) have high probabilities in the *Internet Celebrities* cluster. When a specific NE cluster and its highly related topics are contained in the same news article, they can draw significant attention and make the news article popular. Aside from the above four clusters, there also exist NE clusters, such as *Cities*, that are indistinguishable over all topics.

6. Experimental results

6.1. Overall popularity prediction

Here, the popularity prediction performance for NetEase views, NetEase viewer comments, and Tencent viewer comments, respectively, are presented. The NETM settings for NetEase views are identical to those presented in Section 4.2. For NetEase viewer comments, we set $M_{\max} = 5 \times 10^6$ and the number of topics $K = 160$, while for Tencent viewer comments, we set $M_{\max} = 10^6$ and $K = 100$. The $M_{\max}$ values are chosen based on the range of popularity indicators. Further investigation of the influence of $M_{\max}$ on popularity prediction performance is presented in Section 6.4. Meanwhile, the number of topics K can be further selected using the commonly used measure of perplexity. For illustration purposes, the prediction performance of NETM is only shown under a specific K. In all experiments, the hyper-parameters were selected using a grid search. To ensure reasonable comparison, the number of topics in LDA is made identical to that in NETM.

Table 4 presents the overall prediction results for NetEase views, NetEase viewer comments, and Tencent viewer comments using different methods. In general, the prediction results for different popularity indicators are similar. Specifically, for both RMSE and MAPE, the proposed NETM achieved the smallest values, which indicates its outstanding prediction performance against all competitive methods. As for the other methods, the SVM-NE has the worst prediction performance, as it only uses NEs to characterize the features of the textual content. When more textual features are considered,just as S-LDA, FC, HM, and IDSS have done, the prediction performances improve. LSTM also intends to explore the relationship between web content and popularities but in a more complicated manner. Additionally, the TPAP approach can gain sufficient knowledge from the popularities of correlated topics and similar articles. Both LSTM and TPAP show superior performance to the other competitors, but their performances are still worse than NETM. It may also be noted that the interpretability of LSTM is relatively weak.

Apart from RMSE and MAPE, the prediction accuracy for well-received articles is also considered. As shown in Table 5, different popularity quantiles q are chosen, and the percentages of correctly predicted articles within a specific quantile obtained by different methods are compared. In general, as q increases, the percentages of correctly predicted articles also increase. This is because the popularity prediction task becomes more complex for

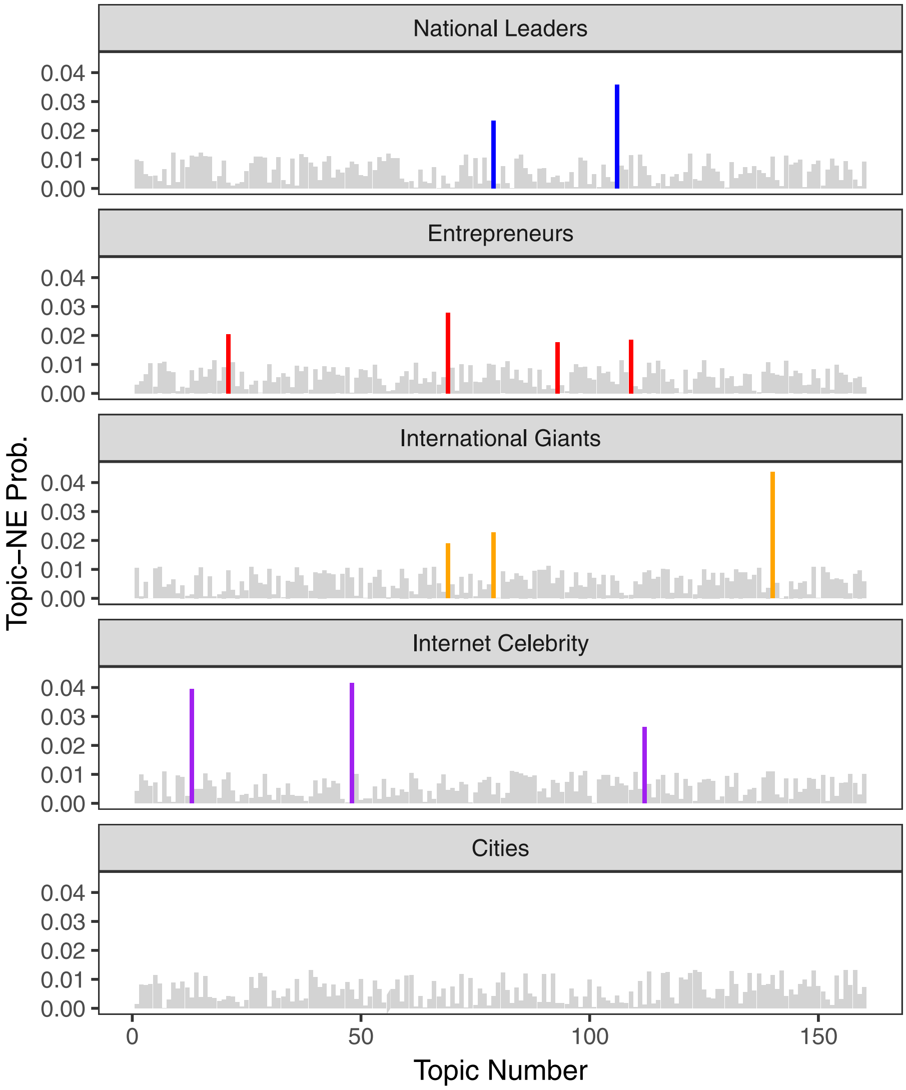


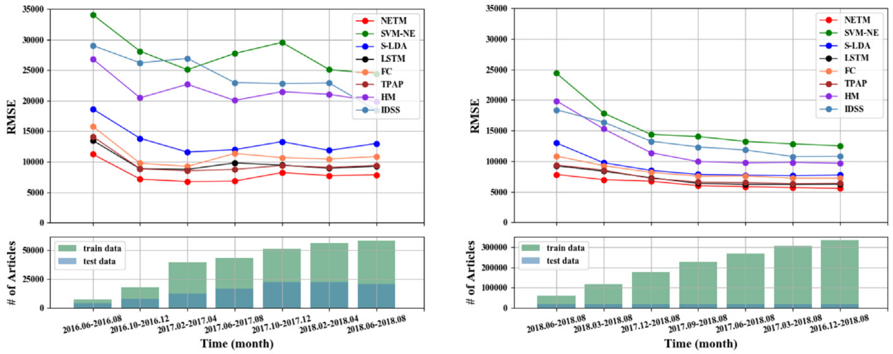
Fig. 3. Popularity distributions of five NE clusters over all topics in the NetEase dataset. The blue topics in the *National Leaders* cluster are *illegal* and *sovereignty dispute*. The red topics in the *Entrepreneurs* cluster are *charity*, *finance*, *crisis* and *debate*. The yellow topics in the *European Football Clubs* cluster are *finance*, *illegal*, and *football competition*. The purple topics in the *Internet Celebrity* cluster are *cosmetic surgery*, *scandal*, and *public relations*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

more popular articles. For different methods, NETM can always achieve the highest percentages of accurately predicted articles, followed by TPAP, LSTM, and other methods. These results verify the strong predictive audibility of NETM. As for different popularity indicators, the Tencent viewer comments achieved higher percentages of accurate predictions than NetEase views and NetEase view comments. This phenomenon is largely due to different distributional patterns of popularity indicators; the Tencent viewer comments are more concentrated, while NetEase views and NetEase viewer comments have more extreme values.

6.2. Prediction along the timeline

This section presents an investigation of the prediction performance of NETM along the timeline. To this end, NetEase views were taken as an example, and two experimental scenarios were considered. In the first scenario, the NetEase published between June 2016 to September 2018 was divided into seven intervals of four months each. Then, the news articles from the first three months and the last month were used as training and testing data, respectively. Fig. 4(a) shows the prediction results of different testing data along the timeline. The associated sample sizes of training and testing data are also present. As shown, the data sizes increase over time. This is because the NetEase website has gained more attention in recent years, and thus, the prediction performances for early time intervals are relatively poor. As for model comparison, the prediction performance achieved by different methods show patterns similar to those seen in Table 4. Specifically, NETM always achieves the optimum prediction performance with the smallest RMSE along the timeline. The LSTM, TPAP, and FC methods also show better prediction performance than the other methods.

In the second scenario, the news articles published in September 2018 were taken as the testing dataset, while a larger training dataset was considered. Specifically, the training dataset was taken as the set of news articles published in the first m months before the prediction time, and let $m = 3, 6, 9, 12, 15, 18$, and 21. For example, the first training dataset lies in the time interval June 2018 to August 2018, while the second training dataset lies between March 2018 to August 2018. Fig. 4(b) shows the prediction results as well as the data sizes along the timeline. In general, the prediction performance of all the methods improved over time, which was primarily due to the enlarged training dataset. However, once the training dataset reached a certain scale, the



(a) First scenario

(b) Second scenario

Fig. 4. Results of predicting the NetEase views along the timeline in two scenarios. The associated sample sizes of training and testing data are also given.

Table 4
Results of popularity prediction using different datasets. The RMSE and MAPE values are calculated over the entire dataset, as well as on the dataset of only non-zero true popularity values (−1) or zero true popularity values (−0).

Measure	NETM	SVM-NE	S-LDA	LSTM	FC	TPAP	HM	IDSS
NetEase Views								
RMSE	7571	24950	15818	8961	11275	8357	24139	24379
RMSE-1	10410	34305	21749	12320	15502	11490	33190	33519
RMSE-0	8.74	31.97	19.64	11.19	14.03	12.13	30.79	29.22
MAPE	3.83	10.94	7.76	4.51	6.03	4.68	10.92	11.24
MAPE-1	0.88	2.01	1.63	1.08	1.58	1.16	3	2.04
MAPE-0	7.14	20.97	14.64	8.36	11.03	8.63	19.81	21.57
NetEase Comments								
RMSE	320	836	572	389	431	383	758	791
RMSE-1	440	1150	787	534	593	527	1041	1088
RMSE-0	8.64	25.19	16.82	9.03	12.27	10.32	23.65	22.06
MAPE	3.79	9.58	6.79	4.07	5.31	5.2	8.73	8.98
MAPE-1	1.19	3.12	2.73	1.26	2.04	3.03	3.67	3.41
MAPE-0	6.71	16.84	11.35	7.23	8.98	7.64	14.41	15.24
Tencent Comments								
RMSE	57	114	81	64	71	66	101	113
RMSE-1	78	154	110	88	97	90	137	153
RMSE-0	8.97	27.96	18.55	10.13	13.46	10.21	25.87	24.79
MAPE	4.53	11.71	8.29	5.03	6.41	5.33	10.05	10.36
MAPE-1	1.16	3.09	2.13	1.47	1.66	1.31	2.73	2.74
MAPE-0	8.32	21.39	15.21	9.03	11.75	9.84	18.27	18.92

Table 5
Prediction accuracy results for well-received articles in different datasets.

Measure	NETM	SVM-NE	S-LDA	LSTM	FC	TPAP	HM	IDSS
NetEase Views								
Top1	47%	21%	28%	43%	41%	47%	21%	21%
Top1–5	52%	23%	30%	45%	42%	52%	22%	22%
Top6–10	54%	24%	31%	49%	45%	53%	26%	26%
Top11–15	55%	26%	34%	50%	47%	54%	30%	30%
Top16–20	59%	29%	36%	53%	48%	56%	34%	34%
NetEase Comments								
Top1	55%	26%	31%	49%	47%	55%	30%	30%
Top1–5	61%	31%	37%	57%	53%	59%	31%	32%
Top6–10	64%	32%	44%	58%	55%	61%	34%	33%
Top11–15	67%	35%	47%	61%	57%	62%	38%	36%
Top16–20	67%	39%	49%	62%	58%	64%	41%	39%
Tencent Comments								
Top1	59%	38%	44%	55%	53%	56%	40%	40%
Top1–5	62%	40%	46%	57%	54%	59%	42%	43%
Top6–10	65%	43%	47%	58%	57%	62%	42%	45%
Top11–15	68%	45%	50%	61%	60%	63%	46%	46%
Top16–20	70%	45%	53%	65%	62%	65%	49%	48%

Table 6
Percentages of top three news categories and the corresponding average number of NEs or NE clusters in NetEase and Tencent, respectively. "Tech.", "Focus.", "Enter." represent "Technology", "Focus News", and "Entertainment".

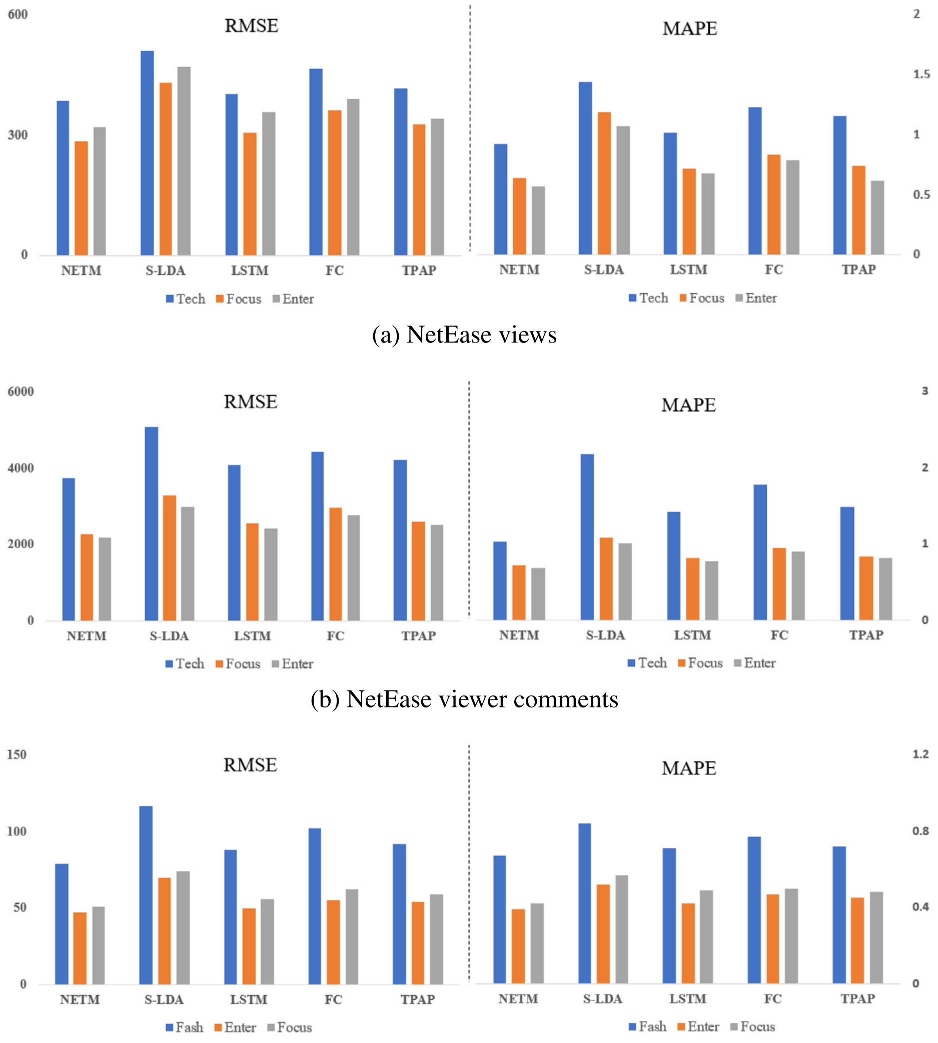
Category	NE				NE Cluster			
	All	People	Place	Org.	All	People	Place	Org.
NetEase								
Tech. (20.7%)	15.45	3.72	5.59	5.14	5.06	2.25	0.74	2.07
Focus (14.8%)	24.73	10.37	8.56	5.80	8.39	3.56	3.42	1.41
Enter. (13.2%)	19.01	9.28	5.37	4.36	7.31	4.12	2.14	1.05
Tencent								
Fashion (18.2%)	12.67	4.86	3.99	3.82	4.99	2.15	1.21	1.63
Focus (17.9%)	18.57	7.53	5.16	5.88	6.31	2.68	1.69	1.94
Tech. (13.1%)	16.71	3.67	6.69	6.35	5.75	2.21	1.12	2.42

increase in data (particularly by adding out-of-date news articles) was of little help to the prediction accuracy. Again, on comparing the different methods, NETM was found to outperform all other models along the timeline.

6.3. Prediction for top categories

The performance of NETM was further investigated in the top three categories in the two datasets. These categories are *Technology*, *Focus News*, and *Entertainment* in NetEase and *Fashion*, *Focus News*, and *Technology* in Tencent, respectively. Table 6 reports the percentages of each category as well as the average number of NEs or NE clusters in each dataset. In both datasets, the category *Focus News* has a higher number of NEs and NE clusters than other categories. In the categories *Focus News*, *Entertainment*, and *Fashion*, names of people rank the highest, followed by locations and organization names. The average number of NE clusters per dataset follows the same pattern. However, in both datasets, the category *Technology* has the largest number of place names, but the smallest number of place clusters. This finding implies that technology news articles generally mention a large number of locations, which have high homogeneity.

Because SVM-NE, HM, and IDSS approaches did not perform well on the entire dataset, they were omitted when handling the prediction task for top categories. Therefore, only NETM, S-LDA, LSTM, FC, and TPAP were applied and compared. Figs. 5(a) and 5(b) show the detailed results for the NetEase dataset, using the number of views and number of viewer comments as popularity indicators. For simplicity, only the results of RMSE and MAPE were reported. In general, the prediction performances of different methods are similar to the overall performances shown in Table 4. Under each of the top three categories, the proposed NETM consistently achieved the smallest RMSE and MAPE values among all prediction results. Among the three categories, the *Focus News* and *Entertainment* categories have relatively smaller



(a) NetEase views

(b) NetEase viewer comments

(c) Tencent viewer comments

Fig. 5. Prediction results for the top three categories, i.e., *Technology*, *Focus News*, and *Entertainment* in the NetEase dataset and *Fashion*, *Focus News* and *Technology* in the Tencent dataset, respectively.

RMSE and MAPE values than *Technology*. This finding may result from the fact that articles in *Focus News* and *Entertainment* categories have more NEs than those in *Technology*. As a result, these two categories can more effectively learn the NE topic probability matrix, and thus, they demonstrate superior popularity prediction performance.

Fig. 5(c) shows the detailed results for the Tencent dataset, using the number of viewer comments as the popularity indicator. Similar to the prediction results for the NetEase dataset, NETM was found to have the optimum prediction performance, measured using both RMSE and MAPE. Among the top three categories, *Entertainment* and *Focus News* once again showed better prediction performance than *Fashion*. This is primarily due to the different number of NEs contained in different categories.

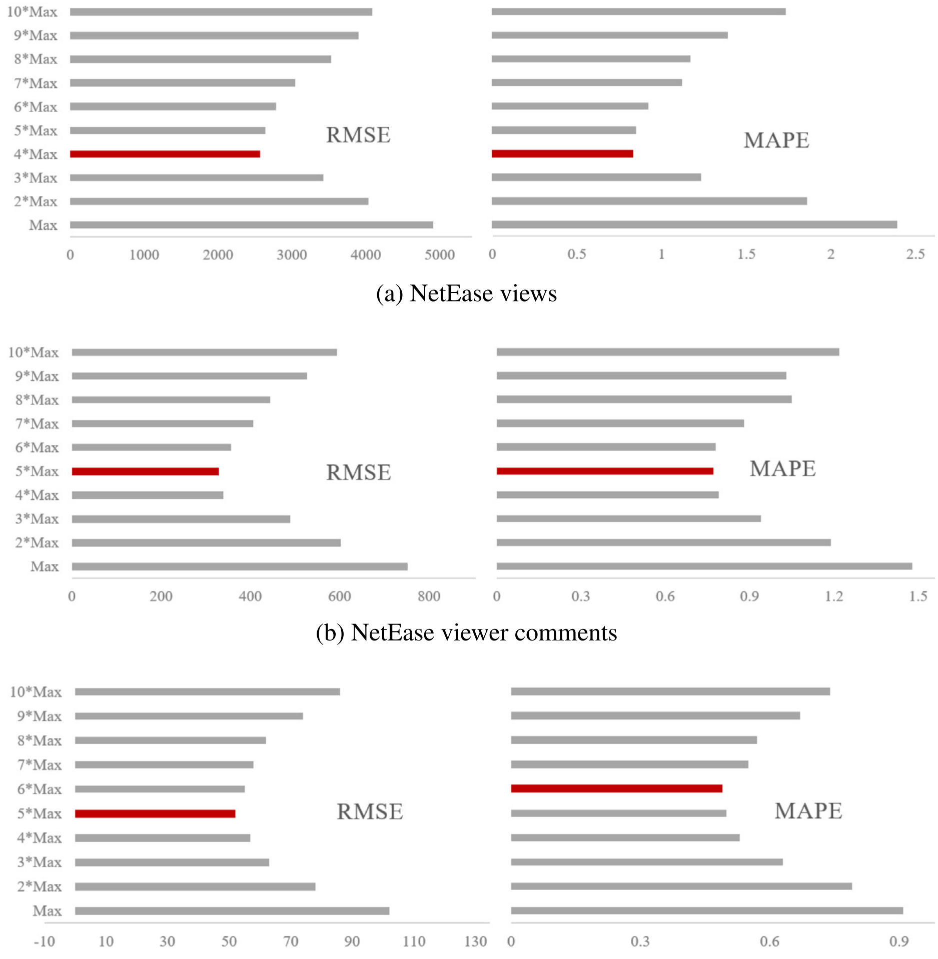
6.4. Investigating the influence of $M_{\max}$

In NETM, the upper limit popularity value $M_{\max}$ must be specified in advance. Therefore, it is necessary to investigate the influence of $M_{\max}$ on the prediction performance. To this end, a series of experiments were conducted using different values of $M_{\max}$. Because $M_{\max}$ represents the upper popularity limit, the maximum value of the popularity indicator (denoted as Max) in the training dataset was taken as the baseline. This was 2,174,184 for NetEase views, 99,617 for NetEase viewer comments, and 40,110 for Tencent viewer comments. Then, several multiples of Max, from Max to 10×Max, were taken as $M_{\max}$, and the corresponding prediction performances of NETM were explored.

Fig. 6 presents the prediction performance under different $M_{\max}$ values for NetEase views, NetEase viewer comments, and Tencent viewer comments, respectively. For illustration purposes, only RMSE and MAPE have been reported. It is clear that as $M_{\max}$ increases, both RMSE and MAPE first decrease and then increase, indicating the existence of inflection points. Specifically, when $M_{\max}$ is relatively small ($<$ 4×Max), it may be insufficient to cover news articles with high popularity values, resulting in poor prediction performance. However, an extremely large $M_{\max}$ value ($>$ 5×Max) would lead to high variance in modeling the popularity indicator, making the model unstable. Therefore, an appropriate $M_{\max}$ is crucial to achieving good popularity prediction performance. As shown in Fig. 6, in the experiments conducted herein, the optimum prediction performance is obtained when $M_{\max}$ is about 5×Max.

7. Model efficiency

To illustrate the computational efficiency of NETM, it is theoretically and empirically compared with the basic LDA model.



(a) NetEase views

(b) NetEase viewer comments

(c) Tencent viewer comments

Fig. 6. Results of RMSE and MAPE under different $M_{\max}$ values in predicting NetEase views, NetEase viewer comments, and Tencent viewer comments. $M_{\max}$ represents the maximum value of the popularity indicator in the training dataset.

Table 7
Time complexity and number of in-memory variables in LDA and NETM.

	Time complexity	Number of in-memory variables
LDA	$O(N_{iter}KD\bar{N})$	$2K(D+V)+D\bar{N}$
NETM	$O(N_{iter}K(V+PD+D+D\bar{N}))$	$2K(D+V)+D\bar{N}+2D+P(D+K)$

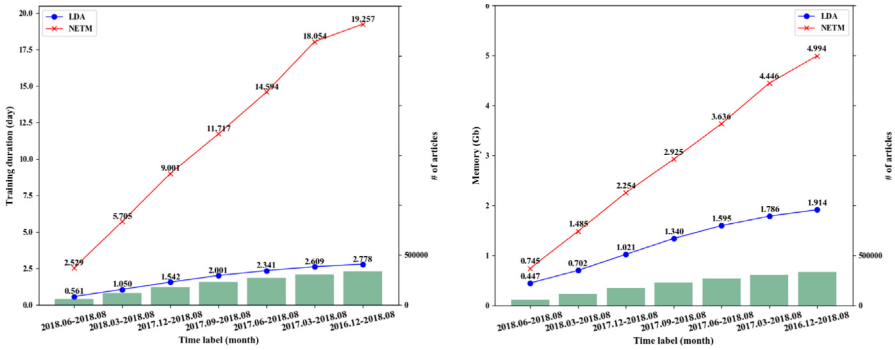
First, the time complexity and memory requirements of both models are shown, following which their running time and memory consumption in experiments under different settings are listed. Both models were trained via Gibbs sampling with a C++ implementation.

7.1. Theoretical comparison

The average length (the number of words) of news articles and the number of iterations in Gibbs sampling are denoted by $\bar{N}$ and N_{iter}, respectively. The time complexity and number of in-memory variables in the Gibbs sampling procedure of the two models are listed in Table 7.

Time complexity. The LDA model draws a topic for each word in the text corpus, with overall time complexity of $O(N_{iter}D\bar{N}K)$. For NETM, there are four sampling steps. In the first step, NETM draws a topic for each word in the text, which is the same as LDA and requires the computational time of the order $O(N_{iter}D\bar{N}K)$. The differences between NETM and LDA lie in the following three steps. The second step calculates word distribution for each topic k over V words, which has a time complexity of $O(N_{iter}KV)$. The third step updates the NE topic matrix with the computational time of the order $O(N_{iter}KPD)$. The final step updates the topic distribution for each document, requiring the computational time of the order $O(N_{iter}DK)$. As a result, the overall time complexity of NETM is $O(N_{iter}K(V+PD+D+D\bar{N}))$. Noting that in a large dataset, we generally have $D < V \ll D\bar{N} < PD$, the time complexity of NETM can be approximated to $O(N_{iter}KD(P+\bar{N}))$. Therefore, the time complexity of NETM is $(P+\bar{N})/\bar{N}$ times the complexity of LDA.

In-memory variables. In the LDA model, count matrices (i.e., $n_{kv}^{(1)}, n_{dk}^{(2)}$) and topic assignments (i.e., z_{dn}) need to be kept in memory in Gibbs sampling iterations. In addition, $\boldsymbol{\phi}_k$ and $\boldsymbol{\theta}_d$ must be computed after model convergence. Hence, the overall required memory size for LDA is $2DK+2VK+D\bar{N}$. In NETM, additional data and variables need to be stored. These include the popularity indicator Y_d and its corresponding mean λ_d, the P-dimensional NE binary vector $\boldsymbol{I}_d$, and the topic-NE distribution $\boldsymbol{\psi}_p$. Therefore, NETM requires additional memory of size $2D+P(D+K)$, which may be very large when P and D are both large. However, as shown in Section 4.2, all NEs were clustered into groups, and NE clusters are taken instead of NEs when constructing the proposed NETM. This can dramatically decrease the number of NEs, and thus efficiently save both computational time and memory space in NETM.



(a) Comparison of training durations

(b) Comparison of memory consumption

Fig. 7. Results of training durations and memory consumption for predicting NetEase views along the timeline.

7.2. Experimental evaluation

To further illustrate the model efficiency of NETM, its time and memory consumption in prediction experiments along the timeline (the second scenario) in Section 6.2 are reported. For comparison, the basic LDA model was also constructed under the same experimental settings, i.e., the same training data, topic numbers, and sampling iterations. All experiments were conducted on a Linux server with Intel Xeon 8176 x 2 central processing units and 512 GB of memory.

Fig. 7 presents the results of time and memory consumption for model training in different experiments. In Fig. 7(a), the NETM and LDA both show an increasing trend in terms of time consumption. However, with increase in the number of articles, NETM requires more computational time than LDA. This is primarily owing to the increasing number of NE clusters used in the model. As suggested by these results, the relationship between the time complexities of NETM and LDA typically follows the theoretical results shown in Table 7, that is, the computational time required by NETM is $(P+\bar{N})/\bar{N}$ times that needed by LDA. For example, in the last time interval, which has 0.35 million articles and 2250 NE clusters, NETM takes a total of 19.725 days, while LDA only takes 2.788 days.

In Fig. 7(b), the memory consumption for NETM and LDA follows a trend similar to that shown in Fig. 7(a), with NETM growing much faster than LDA. It should be noted that in the last two time intervals, the memory consumption of NETM grows significantly faster than the other time intervals. This is because both the number of documents and number of NE clusters grow faster in these two time intervals. Specifically, in the last time interval, NETM requires 4.994 Gbits of memory to hold all variables, while LDA only requires 1.914 Gbits. In summary, these experimental results show that NETM usually requires much larger computational time and memory consumption than the basic LDA model, especially for large datasets. It motivates us to develop parallel algorithm of NETM to increase the computational efficiency in the future work.

8. Conclusion

Web content popularity prediction has a broad range of applications. The high rate of web content propagation requires popularity prediction to be conducted within a short period, or even before publication. Under the circumstance of “cold start", prediction, popularity learning is fairly difficult, as the popularity growth trend is not available. To address this problem, the textual features of the web content are exploited by combining topics and NEs. An NETM is proposed in which popularity is positively correlated with the accumulation of popularity gain of its NEs over all the topics. Extensive experimentation on two real news article datasets demonstrated that NETM can outperform existing approaches in terms of popularity prediction accuracy. The learned NE topic probability matrix can also help in understanding the reasons behind the popularity of certain content, which can further provide insights into public opinion monitoring.

This study has several possible future directions. First, more textual features (e.g., sentiment) can be considered in the NETM to enhance its prediction performance. Second, the joint modeling of NE extraction and NETM is worth consideration. Third, to increase the computational efficiency of NETM, parallel computing algorithms for NETM can be further developed. Finally, there may exist some “cold" NEs that could decrease news popularity. Thus, a more general “NE topic" matrix allowing both positive and negative values can be considered.

CRediT authorship contribution statement

Yang Yang: Conceptualization, Methodology, Resources. **Yang Liu:** Investigation, Software. **Xiaoling Lu:** Formal analysis. **Jin Xu:** Supervision. **Feifei Wang:** Methodology, Writing - original draft, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 72001205, 11971504, 61632002, 61872399, 61872166, 61672264, 61802009), the fund for building world-class universities (disciplines) of Renmin University of China, the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (No. 18XNLG02), the National Key R&D Program of China (No. 2018YFC0830300, 2018YFC0830302), and the Ministry of Education Focus on Humanities and Social Science Research Base (Major Research Plan 17JJD910001). The authors also want to thank Shuai Wang (https://www.cs.uic.edu/~swang/) and Huayi Li (https://www.cs.uic.edu/~hli/) for their meaningful discussions.

References

[1] K. Lerman, T. Hogg, Using a model of social dynamics to predict popularity of news, in: Proceedings of the 19th International Conference on World Wide Web, 2010, pp. 621–630.

[2] R. Bandari, S. Asur, B.A. Huberman, The pulse of news in social media: Forecasting popularity, in: Sixth International AAAI Conference on Weblogs and Social Media, 2012.

[3] A. Tatar, P. Antoniadis, M.D. De Amorim, et al., Ranking news articles based on popularity prediction, in: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, IEEE, 2012, pp. 106–110.

[4] T. Zaman, E.B. Fox, E.T. Bradlow, A bayesian approach for predicting the popularity of tweets, Ann. Appl. Stat. 8 (3) (2014) 1583–1611.

[5] Y. Keneshloo, S. Wang, E.H. Han, et al., Predicting the popularity of news articles, in: Proceedings of the 2016 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, 2016, pp. 441–449.

[6] G. Rizos, S. Papadopoulos, Y. Kompatsiaris, Predicting news popularity by mining online discussions, in: Proceedings of the 25th International Conference Companion on World Wide Web, 2016, pp. 737–742.

[7] H. Pinto, J.M. Almeida, M.A. Gonalves, Using early view patterns to predict the popularity of youtube videos, in: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, 2013, pp. 365–374.

[8] M. Tsagkias, W. Weerkamp, M. De Rijke, Predicting the volume of comments on online news stories, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management, 2009, pp. 1765–1768.

[9] G. Szabo, B.A. Huberman, Predicting the popularity of online content, Commun. ACM 53 (8) (2010) 80–88.

[10] I. Arapakis, B. Cambazoglu, M. Lalmas, On the feasibility of predicting popular news at cold start, J. Assoc. Inform. Sci. Technol. 68 (5) (2017) 1149–1164.

[11] M.T. Uddin, M.J.A. Patwary, T. Ahsan, et al., Predicting the popularity of online news from content metadata, in: 2016 International Conference on Innovations in Science, Engineering and Technology, ICISET, IEEE, 2016, pp. 1–5.

[12] M. Cha, H. Haddadi, F. Benevenuto, et al., Measuring user influence in twitter: The million follower fallacy, in: Fourth International AAAI Conference on Weblogs and Social Media, 2010.

[13] T. Trzciński, P. Rokita, Predicting popularity of online videos using support vector regression, IEEE Trans. Multimed. 19 (11) (2017) 2561–2570.

[14] N. Naveed, T. Gottron, J. Kunegis, et al., Bad news travel fast: A content-based analysis of interestingness on twitter, in: Proceedings of the 3rd International Web Science Conference, 2011, pp. 1–7.

[15] H. Dou, W.X. Zhao, Y. Zhao, et al., Predicting the popularity of online content with knowledge-enhanced neural networks, in: ACM KDD, 2018.

[16] S. Abbar, C. Castillo, A. Sanfilippo, To post or not to post: Using online trends to predict popularity of offline content, in: Proceedings of the 29th on Hypertext and Social Media, 2018, pp. 215–219.

[17] K. Fernandes, P. Vinagre, P. Cortez, A proactive intelligent decision support system for predicting the popularity of online news, in: Portuguese Conference on Artificial Intelligence, Springer, Cham, 2015, pp. 535–546.

[18] F. Gelli, T. Uricchio, M. Bertini, et al., Image popularity prediction in social media using sentiment and context features, in: Proceedings of the 23rd ACM international conference on Multimedia, 2015, pp. 907–910.

[19] A. Piotrkowicz, V. Dimitrova, J. Otterbacher, et al., Headlines matter: Using headlines to predict the popularity of news articles on twitter and facebook, in: Eleventh International AAAI Conference on Web and Social Media, 2017.

[20] Y. Liu, A. Niculescu-Mizil, W. Gryc, Topic-link LDA: joint models of topic and author community, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 665–672.

[21] D. Li, B. He, Y. Ding, et al., Community-based topic modeling for social tagging, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, 2010, pp. 1565–1568.

[22] N. Lin, D. Li, Y. Ding, et al., The dynamic features of Delicious, Flickr, and YouTube, J. Am. Soc. Inf. Sci. Technol. (2012) 139–162.

[23] M. Girvan, M. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. (2002) 7821–7826.

[24] X. Han, L. Sun, An entity-topic model for entity linking, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2012, pp. 105–115.

[25] C. Lin, Y. He, Joint sentiment/topic model for sentiment analysis, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management, 2009, pp. 375–384.

[26] W. Che, Z. Li, T. Liu, Ltp: a chinese language technology platform, in: Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations, Association for Computational Linguistics, 2010, pp. 13–16.

[27] Y. Song, S. Shi, J. Li, et al., Directional skip-gram: Explicitly distinguishing left and right context for word embeddings, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 2 (Short Papers), 2018, pp. 175–180.

[28] J.D. Mcauliffe, D.M. Blei, Supervised topic models, in: Advances in Neural Information Processing Systems, 2008, pp. 121–128.