Prediction of News Popularity via Keywords Extraction and Trends Tracking

Alexander Pugachev\textsuperscript{1(✉)}, Anton Voronov\textsuperscript{2(✉)}, and Ilya Makarov\textsuperscript{1(✉)}

\textsuperscript{1} HSE University, Moscow, Russia
avpugachev@edu.hse.ru, iamakarov@hse.ru
\textsuperscript{2} Sberbank of Russia, Moscow, Russia
voronov.a.dm@sberbank.ru

Abstract. In the last years, news agencies have become more influential in various social groups. At the same time, the media industry starts to monetize online distributed articles with contextual advertising. However, the efficiency of online marketing highly depends on the popularity of news articles. In our work, we present an alternative and effective way for article popularity forecasting with two–step approach: article keywords extraction and keywords-based article popularity prediction. We show the benefits of this technique and compare with widely used methods, such as Text Embeddings and BERT–based methods. Moreover, the work provides an architecture of the model for dynamic keyword tracking trained on the newest dataset of Russian news articles with more than 280k articles and 22k keywords for the popularity of forecasting purposes.

Keywords: Online news popularity forecasting · Keyword extraction · Popularity prediction · BERT · Text embedding

1 Introduction

Since the last century, the way how people find out news has changed significantly. Half a century ago, people discovered news from newspapers, magazines, radio, television, and other media resources. Nowadays, the Internet has become the central resource for consuming information. Today people prefer to receive news about events, politics, sports, economy, and other topics from different digital news sources, online aggregators, and mobile applications. People always want to stay informed and get the news as soon as possible. News agencies publish hundreds of articles daily and continuously keep the reader up to date with events. Today, each person can follow global trends and tendencies in real–time, keep up to date with meaningful events, comment on, and discuss the news that he is interested.

For the last several years, the news agencies’ websites traffic increased significantly\textsuperscript{1}. Any website on the Internet can be considered a platform for

\textsuperscript{1} https://www.liveinternet.ru/stat/lenta.ru/summary.html?lang=en.
advertising, and the news websites are no exception. The higher website traffic provides higher benefits for owners. It is necessary to reduce costs and increase web advertisement efficiency and display ads according to a particular news article’s popularity.

Forecasting news articles’ popularity task has many challenges, mainly because it is often difficult to find a suitable definition and measure popularity. The most appropriate metric for measuring the article’s popularity related to the advertising display is the number of article web page views. From this point of view, it is useful to know in advance will the particular news article become popular and decide whether to show the advertisement on its web page or not.

This paper investigates approaches for solving the news article’s popularity forecasting problem, considering the number of articles views as the base metric. For this purpose, we have collected a dataset consisting of news articles from one of the most famous Russian news agencies, “RIA News”\(^2\) over a long time. We present an article popularity prediction pipeline, which consists of two steps: relevant keywords extraction and keywords–based popularity prediction. We compare this approach to the most popular ones based on text embeddings and BERT \([4]\) model.

The paper is structured as follows. Section 2 presents an overview of related works; Sect. 3 introduces the dataset; Sect. 4 describes the proposed method. The experiment results present in Sect. 5. Finally, Sect. 6 summarizes the conclusion.

2 Related Work

Although news popularity prediction is a relatively novel problem, it is in many researchers’ scope of interest. Various studies consider different ways to measure news article popularity and related data to make predictions.

Balali et al. \([2]\) took the number of comments as the base metric of the article’s popularity. For building a reliable predictive model, authors extracted both textual (article title, body, comment tree) and non–textual (number of views, likes, dislikes) information from article web pages. Lee et al. \([9]\) proposed an unsupervised keyword extraction technique that can be used for tracking news topics over time. Authors introduced six weighted TF–IDF variants and applied keyword extraction for several news domains (e.g., politics, business) separately. Keneshloo et al. \([7]\) took into account that the life span of a news article is relatively short. It is more useful and valuable to predict an article’s early popularity rather than its long-term popularity. Their study authors defined an article’s popularity as the number of page views within the first 24 h after publication. The proposed model tracks the article for 30 min upon publication, extracts a range of temporal, social, and contextual features, and makes accurate forecasting. Xia et al. \([18]\) analyzed base roles such as people, organizations, and political parties to understand the main trends in the news. Gayberi et al. \([5]\) considered

\(^2\) https://ria.ru/.
the task of popularity prediction of posts in social networks taking into account user, post and statistical features, and image object detection related features.

Piotrkowicz et al. [14] thoroughly studied the wide variety of textual features (morphological, semantic, syntactic, etc.) of articles’ titles and built a reliable predictive model based on a support vector machine. Lamprinidis et al. [8] considered multi–task learning approach for news headlines popularity prediction. The main objective for researchers was to build a machine learning model based on the title of a news article that predicts whether users will click on this title or not. Lu et al. [10] performed an analysis of people’s behavior when reading online news articles. They found that users are more likely to click on low–quality articles because of their high title persuasion. Voronov et al. [16] developed a language–independent model with an Online Machine Learning approach. Researchers trained and evaluated models on Russian and Chinese articles considering different text preprocessing methods according to these languages’ morphology and syntax.

Some researches preferred to use sentiment analysis to predict the popularity of the content. In the work of Wang et al. [17] the popularity indicates by the sentiment analysis of the popular internet trends. However, it can predict only the class of the sentiment for most observed users, but it can not predict the final number of views. The model can indicate a person’s attitude to a particular topic, but not the retention of the article in the news feed.

In contrast to the previous works, we consider articles’ relevant keywords as the basis for making predictions. Using keywords, we can obtain a concise representation of a particular news article and track trends’ dynamics.

3 Dataset

3.1 Data Collection

There exist many news articles datasets publicly available on the Internet, the majority of them are English–language. One of the most appropriate for our study Russian–language dataset is “News dataset from Lenta.Ru”3. However, it does not contain any information about articles’ number of views. Since no other dataset with the latest news and sufficient for models training number of samples, it was decided to collect a suitable dataset on our own with the “RIA News” news agency as the base source of news articles. “RIA News” is one of the world’s largest news agencies, according to LiveInternet, it has more than 62.9 million visitors per month4. We took articles for the period from May 2006 to March 2020.

3.2 Dataset Information

Each of the articles has a title, body, a set of relevant keywords, information about its publication time, and views count. An element from the collected

3 https://www.kaggle.com/yutkin/corpus-of-russian-news-articles-from-lenta.
4 https://www.liveinternet.ru/stat/RS_Total/Riaru_Total/summary.html.
corpora represents a JSON object. An example from the dataset presented in Table 1, all the fields have been translated from Russian to English.

Table 1. Example of one item from the collected dataset.

| Key | Value |
|-----|-------|
| Title | It became known who Meghan Markle wants to play when she returns to the movies |
| Link | https://ria.ru/20200302/1566826292.html |
| Time | 21:53 02.03.2020 |
| Topic | World |
| Views | 2108 |
| Article | The media discovered that the Duchess of Sussex Meghan Markle asked her agent nick Collins to find her a role in the Marvel superhero blockbuster, the Mirror reports. The source reports that Prince Harry’s wife is confident that participating in such a major project will help her revive her acting career... |
| Tags | Culture, Movies and TV series, Culture news, Meghan Markle |

There are 23,254 unique keywords in the collected dataset. The keywords distribution (sorted by descending order of occurrence) among the articles as well as the distribution of the number with respect of unique keywords for each article shown in Figs. 1 and 2 respectively. Both histograms are presented on a logarithmic scale. Approximately 98% of articles have less than 8 related tags.

![Fig. 1. Keywords distribution among all articles.](image)

![Fig. 2. Distribution of number of article’s keywords.](image)

The distribution of the number of views and the distribution of articles’ lengths in terms of the number of words are shown in logarithmic scale on Figs. 3 and 4 respectively.
3.3 Data Preprocessing

The data preprocessing procedure included deleting duplicate articles, items with no title, text, or set of tags, and samples that do not have a time of publication or number of views. After the preprocessing step, the number of articles decreased from approximately 335k to 296k.

The keywords preprocessing is also worth mentioning. Firstly, several most popular tags were removed from each of the articles. Many keywords are close semantically in the collected dataset and are similar in meaning but different in spelling. Several examples of these cases provide in Table 2, the names of the keywords are translated from Russian to English.

| Keyword #1                           | Keyword #2                           |
|--------------------------------------|--------------------------------------|
| Analytics—Religion and worldview     | News – Religion and worldview        |
| G20 Summit in Argentina              | G20 Summit                            |
| Khanty–Mansi Autonomous Okrug        | Khanty–Mansi AO                       |
| Space                                | Space—RIA Science                    |

To reduce the number of unique keywords by replacing a pair of similar keywords with a single one, we took the most frequently encountered keyword pairs and using Yandex. Toloka\(^5\) obtained a labeled dataset of keyword pairs. The assessors marked whether a suggested pair of keywords are similar in meaning or not. If the majority of assessors marked two keywords as similar, then these keywords were merged into one. There were involved approximately 350 assessors in keyword similarity labeling. After merging similar keywords, the number of unique tags among all articles decreased from 23,254 to 22,919.

\(^5\) https://toloka.yandex.com/.
4 Methodology

The popularity of a specific news article directly depends on the topic of the article. A set of relevant keywords often describes one specific topic. Therefore, based on keywords, the article’s topic can be determined and predicted. The proposed model involves first extracting relevant keywords from a news article and then predicting the number of views based on the predicted keywords.

4.1 Keyword Extraction

For the keyword extraction pipeline, we combined the One–Shot Learning approach for building a custom vectorizer and k–Nearest Neighbors multi–label classification algorithm for obtaining relevant keywords.

Vectorizer Model. Despite the existence of several well-established text embedding methods such as Word2Vec [11], FastText [6], and ELMo [13], we implemented our vectorizer model taking into consideration the specificity of our task and data. Before treating a text sample to the model, it goes through the tokenization procedure the same way as when working with the BERT model. The RuBERT\(^6\) vocabulary used for tokenizing.

We built a Siamese Neural Network, which produces multi–dimensional embedding vectors close to each other for two similar texts. For dissimilar samples, it produces two vectors with a large distance between them. The model particularly takes input a triplet of tokenized texts: an Anchor article, one that is similar to it (called Positive), and another one that is dissimilar to it (called Negative). Before training, all the articles were sorted in ascending order by publication time (oldest articles were in the beginning). While training for a particular Anchor article, there were randomly taken one Positive and one Negative article among 2000 closest by date of published articles. The rule for determining the similarity between a pair of articles was the following:

- Two articles are considered \textit{similar} if they have no less than a half common keywords
- Two articles are considered \textit{dissimilar} if they have strictly less than a half common keywords

The key feature of such architecture consists in specific loss called Triplet Loss [15]:

$$\mathcal{L}(s_a, s_p, s_n) = \max \{ ||s_a - s_p|| - ||s_a - s_n|| + \alpha, 0 \}$$  \hspace{1cm} (1)

where \(s_a, s_p, s_n\) are vector embeddings for Anchor, Positive and Negative samples, respectively, \(|| \cdot ||\) is Euclidean distance and \(\alpha\) is hyperparameter (was set to 0 for all experiments). The architecture of vectorizer model is depicted at Fig. 5.

\(^6\) http://docs.deeppavlov.ai/en/master/features/models/bert.html.
Multi-label kNN. In order to predict a set of relevant keywords for a specific article, we construct a multi-dimensional vector space from embeddings of previously published articles. There are found $k$ its nearest neighbors for the article according to cosine similarity metrics, the Top-$n$ most frequent keywords among these neighbors considered as predicted keywords for this article. The value 25 for the $k$ parameter and value 3 for the $n$ parameter were found to be optimal during experiments.

4.2 Popularity Forecasting

To predict the popularity of a news article regarding the number of views based on the relevant keywords, we applied the Tracking Window strategy with gradient boosting.

Tracking Window. For a certain moment of time $t$ the popularity of a keyword can be calculated as follows:

$$P_{(t,K)} = \frac{\sum V_{d_t \cap K}}{N_{d_t \cap K}}$$

where $P_{(t,K)}$ is the popularity of keyword $K$ at the moment $t$, $\sum V_{d_t \cap K}$ is the sum of all views of articles for which keyword $K$ is relevant for the time moment $t$, and $N_{d_t \cap K}$ is a number of articles for the moment $t$ for which keyword $K$ is relevant.

During the experiments, one day was taken as a time moment $t$. To predict the impact of a specific keyword’s popularity on the number of views, we consider the keyword’s tendencies for $N$ past moments using the Tracking Window mechanism.

The main steps of processing articles’ keywords into a Tracking Window matrix shows in Fig. 6. First, for each article’s keyword, calculates its average popularity for the past $N$ days. The Tracking Window ($N$ parameter) was set to 30 (days), including the day of publication of a specific article, which allows determining general and local trends more accurately. After that, the Top–3
article’s keywords were taken according to the average popularity and merged into a $3 \times N$ matrix. The obtained matrix represents the trends of each keyword individually and, at the same time, shows the overall trend for a news article. If a specific article had less than 3 keywords, the matrix’s corresponding values for missing keywords were set to 0 for each of the $N$ days.

**Number of Views Forecasting.** Gradient boosting was chosen as a model for predicting the number of views due to its ability to analyze data sequences and tendencies, which are reflected in the Tracking Window. Moreover, the approach itself implies the continuous model actualization according to changing trends or news resource. Algorithms based on decision trees, in particular gradient boosting, are well suited for such requirements. We used one of the most widely used implementations of gradient boosting through our study, CatBoost [12].

Before treating the Tracking Window matrix to CatBoost, it was converted to a one-dimensional vector. According to the Top–3 keywords and separated by a “−1” value, each of the rows was situated. The obtained vector was treated to the CatBoost regression model, which outputs the number of views. The full forecasting pipeline for obtaining the number of views based on the Tracking Window matrix is presented in Fig. 7.

### 4.3 Full Prediction Pipeline

The proposed popularity prediction pipeline has presented in Fig. 8. Firstly the article’s vector embedding is obtained using a trained vectorizer model. Then, the kNN multi–label classifier used for extracting 3 most relevant keywords. Finally, considering predicted keywords and news articles for the past 30 days, the
Fig. 7. Number of views forecasting based on Tracking Window. The Tracking Window matrix is turned into a one–dimensional vector and then treated to CatBoost Regression model which outputs the number of views.

Tracking Window matrix is built and treated to the trained CatBoost Regressor model for predicting views.

Fig. 8. Full popularity forecasting pipeline.

4.4 Comparable Approaches

To show the keyword extraction step’s usefulness, we compare our approach to the most popular ones nowadays. We considered the BERT model and a neural network with trainable custom embeddings and two fully connected layers with Dropout. These models took as input tokenized articles and directly predicted the number of views without considering relevant keywords.
Regarding BERT, we considered a RuBERT pre-trained model with 128-unit Feed Forward layer with ReLu [1] activation function before the model’s output. During the training, the RuBERT model was frozen in the case of the original BERT model issue, which has a limit of quality for the regression task⁷.

5 Experiments and Results

We evaluated keyword extraction and views prediction models, including the proposed one. The training set consisted of news articles from May 2006 to November 2019; the test set included articles from November 2019 to March 2020. During training, both vectorizer and the views prediction models training samples were treated consequently according to the articles’ publication date without any shuffling.

5.1 Metrics

We chose F1 score as a metric for keyword extraction evaluation. Regarding views prediction models, the Mean Absolute Error (MAE) metric was used during each of them’ training. For the views prediction evaluation, we took Root Mean Squared Logarithmic Error (RMSLE) [3]. This metric is considered robust to outliers and penalizes the underestimation more severely than the overestimation. Also, we evaluated models in terms of percentile rank concerning Absolute Logarithmic Error.

Table 3. Keyword prediction performance.

| ID | kNN space set     | Text               | Similar keywords | F1  |
|----|-------------------|--------------------|------------------|-----|
| 1  | May 2006—Nov 2019 | Full article       | Merged           | 0.445 |
| 2  |                   |                    | Not merged       | 0.435 |
| 3  |                   | First paragraph    | Merged           | 0.439 |
| 4  |                   |                    | Not merged       | 0.431 |
| 5  | Jan 2019—Nov 2019 | Full article       | Merged           | **0.465** |
| 6  |                   |                    | Not merged       | 0.452 |
| 7  |                   | First paragraph    | Merged           | 0.455 |
| 8  |                   |                    | Not merged       | 0.446 |

⁷ https://github.com/google-research/bert/issues/462.
5.2 Keyword Extraction

The most critical information of the news article concentrates on its first paragraph. Hence, we conducted several experiments where instead of the full articles’ text, we took only their first paragraphs. Also, we evaluated keyword prediction depending on the set of articles in the kNN space. Two sets were considered: from May 2006 to November 2019 and from January 2019 to November 2019. Finally, we trained and evaluated the keywords prediction model before and after merging similar keywords.

Results for keyword extraction are presented in Table 3. It can be clearly noticed that after merging semantically close keywords, the overall performance increased. We can also see that the number of articles located in the kNN space plays a significant role. Using only the most recent articles, instead of all ever published, the keyword prediction performance gets better according to F1 score. Ultimately, in Table 3, we can notice that the quality of predictions using the first paragraphs is not significantly lower compared to the use of full articles’ texts. The hypothesis confirms that the articles’ first paragraphs contain the most important information for prediction.

![Distribution of the number of correctly predicted keywords](image)

Fig. 9. Distribution of the number of correctly predicted keywords. IDs correspond to models in Table 3.

For each keyword prediction model we present the fraction of correctly predicted keywords, shown in Fig. 9. According to the diagrams, with a probability of 90%, our best model predicts at least 1 of 3 keywords correctly. With probability, more than 45% at least 2 of 3 keywords are predicted correctly.
5.3 Popularity Forecasting

We conducted two experiments on the popularity forecasting evaluation for the proposed approach: based on predicted and ground truth keywords. In the first case, for each article from the training set, we predict keywords based on previous articles for the last 2 months and train the CatBoost Regressor model for predicted keywords. In the second case, we took ground truth keywords for each article from the training set. The evaluation results in terms of a percentile rank and RMSLE metric for each of the mentioned models are presented in Fig. 10 and Table 4, respectively. We provide some examples of views predictions in Table 5.

![Fig. 10. Evaluation of number of views prediction in terms of percentile rank with respect to Absolute Logarithmic Error.](image)

**Table 4.** Models evaluation in terms of Root Mean Squared Logarithmic Error.

| Method                          | Root Mean Squared Logarithmic Error |
|---------------------------------|------------------------------------|
| RuBERT                          | 2.019                               |
| Custom embeddings               | 1.708                               |
| Predicted keywords + CatBoost   | 1.672                               |
| Ground truth keywords + CatBoost| 1.042                               |

From Table 5 we can see that our model behaves differently depending on the article’s topic and trends preceding its publication. For instance, examples 2, 3, and 9 show that in cases where news is preceded by related events, predictions are made more accurately. However, in cases when an event happens suddenly, we can get both high-quality (examples 1, 4) and low-quality predictions (examples 5, 7).
Table 5. Number of views prediction examples.

| #  | Article’s title                                                                 | Ground truth number of views | Predicted number of views | RuBERT | Custom embeddings | CatBoost | Predicted keywords | Ground truth keywords |
|----|--------------------------------------------------------------------------------|------------------------------|---------------------------|--------|-------------------|----------|-------------------|----------------------|
| 1  | In Sydney, a group of baboons escaped from the Royal hospital.\(^a\)              | 623                          | 3215                       | 1643   | 1584              | 583      |                   |                      |
| 2  | The Samsung S10’s price reached minimum on the eve of the new products’ release.\(^b\) | 6865                         | 1602                       | 2899   | 1792              | 7029     |                   |                      |
| 3  | The Ministry of Health gave recommendations on the prevention of coronavirus.\(^c\) | 11205                        | 1282                       | 1885   | 3701              | 11265    |                   |                      |
| 4  | Aeroflot received a plane with beds and a bar.\(^d\)                             | 303                          | 2627                       | 1759   | 739               | 214      |                   |                      |
| 5  | Magnitude 5.5 earthquake jolts Japan.\(^e\)                                     | 1675                         | 2456                       | 504    | 812               | 879      |                   |                      |
| 6  | Boeing did not receive orders for planes in January for the first time since 1962.\(^f\) | 2126                         | 1262                       | 3329   | 4566              | 2053     |                   |                      |
| 7  | The neural network colored the video of 1896 about Moscow.\(^g\)                 | 10502                        | 5778                       | 7115   | 2422              | 4786     |                   |                      |
| 8  | Ukrainian citizen refused to evacuate from China without her dog.\(^h\)         | 6439                         | 4705                       | 5539   | 3564              | 20158    |                   |                      |
| 9  | A Serbian pensioner went on foot to Moscow for the parade on May 9.\(^i\)        | 9142                         | 3725                       | 2360   | 4051              | 11289    |                   |                      |
| 10 | Jack Ma donated money for the development of remedy against coronavirus.\(^j\)   | 783                          | 775                        | 918    | 1582              | 2134     |                   |                      |

\(^a\)https://ria.ru/20200225/1565191567.html
\(^b\)https://ria.ru/20200212/1564602625.html
\(^c\)https://ria.ru/20200130/1564059279.html
\(^d\)https://ria.ru/20200304/1568142823.html
\(^e\)https://ria.ru/20200212/1564594675.html
\(^f\)https://ria.ru/20200211/1564568950.html
\(^g\)https://ria.ru/20200228/1565339495.html
\(^h\)https://ria.ru/20200220/1565632131.html
\(^i\)https://ria.ru/20200216/1564846656.html
\(^j\)https://ria.ru/20200214/1564787083.html

6 Conclusion

This paper presents an approach for predicting news articles’ popularity based on articles’ keywords. This work shows that the keyword-based prediction is a more flexible tool than the text-based prediction. Using keywords as a base feature, we can estimate the real history of trends’ popularity and make better predictions. Experiments show that the prediction of articles’ views based on keywords gives the maximum quality on the test dataset. Moreover, the proposed model has
potential for improvement: due to limited resources, we trained CatBoost on 10
million iterations, the model did not finish training while the other models met
at a certain minimum. The most promising improvement, in our opinion, is to
reduce the number of unique keywords, which can lead to significant growth in
quality. It is worth noting that the proposed model shows acceptable quality
up to 80% of cases.

This work also shows that the deep linguistic model does not show good
quality in this task because news articles already have a compressed semantic
representation, which indicates certain words’ frequencies, which are, to some
extent, a representation of keywords. However, a large pre-trained linguistic
model is tied to the search for linguistic meaning for a specific time moment,
which could be useless for predicting articles’ popularity.

References
1. Agarap, A.F.: Deep learning using rectified linear units (ReLU). CoRR
   abs/1803.08375 (2018)
2. Balali, A., Asadpour, M., Fajil, H.: A supervised method to predict the popularity
   of news articles. Computación y Sistemas 21, 703–716 (2018)
3. Desai, K.V., Ranjan, R.: Insights from the Wikipedia contest (IEEE contest for
data mining 2011). arXiv preprint arXiv:1405.7393 (2014)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirec-
tional transformers for language understanding. arXiv preprint arXiv:1810.04805
   (2018)
5. Gayberi, M., Oguducu, S.G.: Popularity prediction of posts in social networks based
   on user, post and image features. In: Proceedings of the 11th International Confer-
ence on Management of Digital EcoSystems, MEDES 2019, pp. 9–15. Association
   for Computing Machinery, New York (2019)
6. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text
classification. CoRR abs/1607.01759 (2016)
7. Keneshloo, Y., Wang, S., Han, E.H., Ramakrishnan, N.: Predicting the popularity
   of news articles. In: SDM (2016)
8. Lamprinidis, S., Hardt, D., Hovy, D.: Predicting news headline popularity with
   syntactic and semantic knowledge using multi-task learning. In: Proceedings of the
2018 Conference on Empirical Methods in Natural Language Processing, Brus-
sels, Belgium, pp. 659–664. Association for Computational Linguistics, October–
November 2018
9. Lee, S., Kim, H.J.: News keyword extraction for topic tracking. In: 2008 Fourth
   International Conference on Networked Computing and Advanced Information
   Management, pp. 554–559. October 2008
10. Lu, H., Zhang, M., Ma, W., Shao, Y., Liu, Y., Ma, S.: Quality effects on user
preferences and behaviors in mobile news streaming. In: The World Wide Web
   Conference, WWW 2019, pp. 1187–1197. Association for Computing Machinery,
   New York (2019)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word represen-
tations in vector space. CoRR abs/1301.3781 (2013)
12. Ostroumova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: CatBoost:
   unbiased boosting with categorical features. arXiv preprint arXiv:1706.09516
   (2018)
13. Peters, M., et al.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2018)

14. Piotrkowicz, A., Dimitrova, V., Otterbacher, J., Markert, K.: Headlines matter: using headlines to predict the popularity of news articles on Twitter and Facebook. In: ICWSM, pp. 656–659, May 2017

15. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823, June 2015

16. Voronov, A., Shen, Y., Mondal, P.K.: Forecasting popularity of news article by title analyzing with BN-LSTM network. In: Proceedings of the 2019 International Conference on Data Mining and Machine Learning, ICDMML 2019, pp. 19–27. Association for Computing Machinery, New York (2019)

17. Wang, C., Xiao, Z., Liu, Y., Xu, Y., Zhou, A., Zhang, K.: SentiView: sentiment analysis and visualization for internet popular topics. IEEE Trans. Hum. Mach. Syst. 43(6), 620–630 (2013)

18. Xia, C., Zhang, H., Moghtader, J., Wu, A., Chang, K.W.: Visualizing trends of key roles in news articles. arXiv preprint arXiv:1909.05449 (2019)