International Journal of Digital Crime and Forensics
Volume 12 • Issue 4 • October-December 2020

A Light Recommendation Algorithm of We-Media Articles Based on Content

Xin Zheng, Institute of Computing Technology, Chinese Academy of Science, China
Jun Li, Institute of Computing Technology, Chinese Academy of Science, China
Qingrong Wu, Institute of Computing Technology, Chinese Academy of Science, China

ABSTRACT

Since the explosive growth of we-medias today, personalized recommendation is playing an increasingly important role to help users to find their target articles in vast amounts of data. Deep learning, on the other hand, has shown good results in image processing, computer vision, natural language processing, and other fields. But it's a relative blank in the application of we-media articles recommendation. Combining the new features of we-media articles, this paper puts forward a recommendation algorithm of we-media articles based on topic model, Latent Dirichlet Allocation (LDA), and deep learning algorithm, Recurrent Neural Networks (RNNs). Experiments on the real datasets show that the combined method outperforms the traditional collaborative filtering recommendation and non-personalized recommendation method.

KEYWORDS

LDA, LSTM, Recommendation, RNNs, We-Media Articles

1. INTRODUCTION

The application Scenario of this paper is an APP called "Wei-Mi". It focuses on recommending we-media contents of WeChat, WeiBo, ZhiHu to users. The mainly topic of this paper is we-media article recommendation.

Along with the development of Internet, we-medias such as WeChat official accounts have been developed very well. We-media is becoming one of the main information sources of the public. According to the interim results report of Tencent released in 2016, WeChat's monthly active users (MAU) has reached 806 million. Number of WeChat official accounts has reached more than 12 million. There are tens of millions of WeChat official accounts (Penguin intelligence, 2015), and they produce a lot of articles every day. It's hard for users to find their target articles in vast amounts of consultations just rely on themselves. A recommendation system is needed to solve this problem.

Traditional article recommendation algorithms like Collaborative filtering recommendation (Rush A M, 2015), Content based recommendation (Werbos P J., 1990) and Most popular recommendation. The Collaborative filtering recommendation is combined with user-based collaborative filtering and item-based collaborative filtering. Generally, article recommendation is based on user-based collaborative filtering, which is to get a user set S that is similar to the target user u and recommend

DOI: 10.4018/IJDCF.2020100106

what do users in the set S like but user u don't know yet. The Content-based recommendation is to recommends new articles that are similar to articles the user liked before.

The Most-popular recommendations are typically based on the popularity of articles (read number, like number). This algorithm recommends popular articles to users.

There are many kinds of article recommendations system, such as a news recommendation system, scientific literature recommendation system and E-mail recommendation system (Rush A M, 2015). But We-media users show some new characteristics when they read. Firstly, users tend to read more personalized articles compared with the news recommendation which pay more attention to holding popularity and timeliness of news. Secondly, we-media reading shows new characteristics of fragmentation reading and speed reading. Thirdly, according to statistics, most of the We-media users tends to read the latest articles. But there are not enough historical data on new articles.

In this paper, to cover these new characteristics, we put forward a recommendation algorithm of we-media articles based on topic model, Latent Dirichlet Allocation (LDA), and deep learning algorithm, Recurrent Neural Networks (RNNs).

This paper starting from the simple intuitive notion of preserving information. Section 3 introduce the basics knowledge of this paper. To analyze the reading characteristics of We-media users, we also used actual data to measure the user's reading behavior in section 2. Section 3 shows the LDA model (Blei D M, 2003) and the traditional LDA article recommendation algorithm (XiangLiang., 2016), the RNNs algorith (Medsker L R, 2001) and its improved version LSTM algorithm (Sundermeyer M, 2012). All of those lead to Section 5 to motivate the LDA-LSTM recommendation algorithm. Section 4 describes how the LDA–LSTM recommendation algorithm is implemented in the We-media article recommendation system. This section expounds the structure of the LDA-LSTM algorithm and describes the training method. It presents experiments that show considerably outperforms of LDA-LSTM algorithm. We also show experimental results of Random recommendation, Most-popular recommendation and Collaborative-filtering recommendation (Linden G, 2003). The result shows that the LDA-LSTM algorithm has obvious improvement on the precision and recall rate compare with 3 other algorithms. And it performs better than the other three algorithms on new articles. Section 6 summarizes our findings and describe the outlook of our work.

2. WE-MEDIA USERS' CHARACTERISTICS ANALYSIS

2.1. Data Set

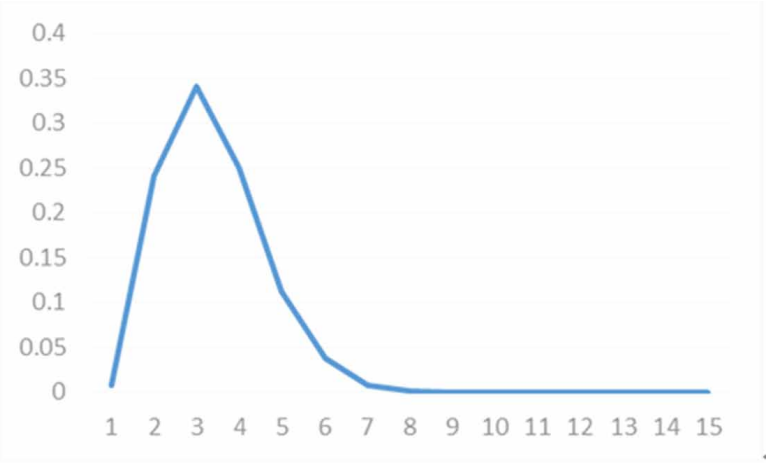
To fit the We-media users' reading characteristics better, this section analysis users' reading habits according to the actual log data analysis. This article collects the behavior logs of 10, 000 users from March 31, 2016 to March 31, 2016.

2.2. Personalized Characteristic of We-Media Users

In this section, we firstly extracted 15 topics distribution from articles with the LDA algorithm. Secondly, based on users' history reading behavior, we count up of the number of topic that users are interested in. The picture in Figure 1 presents the statistical results, where the horizontal coordinates represent the number of topics of interest, and the ordinate indicates the proportion of users.

The more personalized the user profile, the more left the expected value will be, and the more fragmented the user will be, the more likely the expected value will be to the right. The picture presents the expected value of topics that the media users focus on is more left-leaning, so the personalized features of the We-media users are obvious. This suggests that the emphasis and timeliness of the traditional news recommendation system are more than individualized, and that the recommendation of media content should emphasize personalization.

Figure 1. Statistical results of users' interest



2.3. Fragmentation Reading and Speed-Reading Characteristic of We-Media Users

The picture in Figure 2 presents how long the reading time of We-media user takes. 76 percent of users' reading time is less than 10 minutes and for 10 minutes to 30 minutes is 13 percent. The picture in Figure 3 presents statistics of interval time of users' reading. 75% of time difference of two reading behavior is up to 8 hours. Proportion of 4 to 8 hours is 14%. Fragmentation and speed-reading feature are quite obvious.

By observing the log, we found that users generated a series of clicks in a short period of time. And the sequences of the users' clicking to are not isolated. The current clicking behavior is affected by the pre-behavior of this sequence. The current reading articles will also affect users' interest at the time and affect the decision-making. The picture in Figure 4 presents the user "9106148" read an article about "Liu han's rise and fall survey series of four" which inspired his interest in at the

Figure 2. Statistics of users' reading time

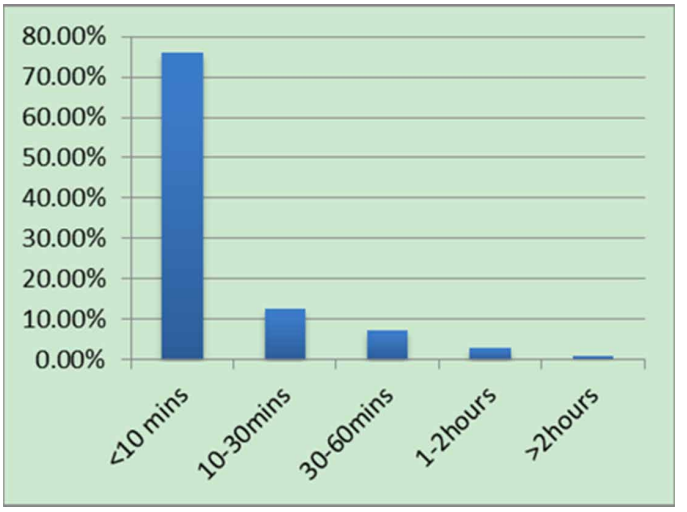
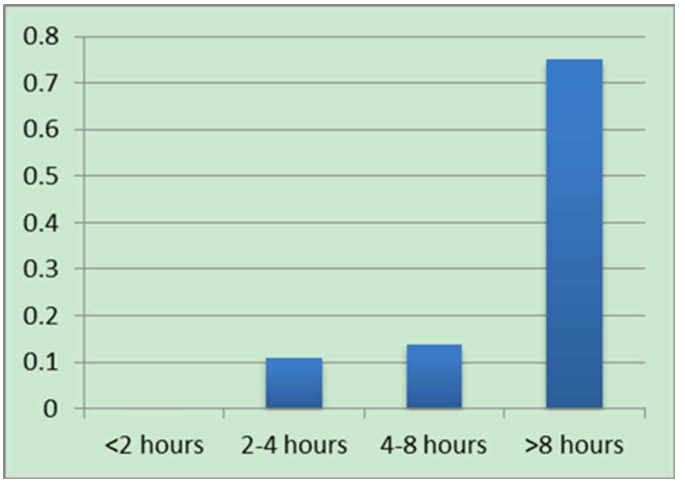


Figure 3. Statistics of time difference of 2 reading



moment, then the user read several other related articles of this series, in 2015 March 15 16:22:03. It takes about 14 minutes from stimulating interests to finally finished reading.

We-media users always browse articles in pieces of time such as taking a subway, having a cup of coffee. The characteristics of fragmentation reading and speed reading make users' reading habits much affected by environmental factors and mood. Traditional recommendation method based on the user long-term historical behavior is difficult to meet this feature. Thus, what we need is an algorithm that is based on the user's short-term behavior, to capture user current interest changes better and mining user reading sequence contains rules.

2.4. Users Tend to Read New Articles

The picture in Figure 5 presents that this article counted the time difference of reading time and delivery time of the We-media article. The abscissa is a unit of time is hours, ordinate is the number of logs on the time difference. We can see that the majority of users' reading time is within 28 hours of publication, accounting for 96.2% of users.

There are not enough historical data on new articles to train model which leads to algorithms like collaborative filtering have a poor performance. Content-based recommendation algorithms can mitigate this problem.

3. LDA-LSTM RECOMMENDATION ALGORITHM

3.1. Algorithm Architecture

Assume $\vec{W}_i = (1, 0.....,1)$ is one-hot coding of article i for each topic. Users' visiting sequences can be transformed to a series of $\vec{W}_i$ vectors. The picture in Figure 6 presents those vectors to binary numbers and let them be the input-data of LSTM model.

With the constant of input, LSTM model forecasts the topic-distribution of next moment. It foresees user's mood and interest next time, so we assign it to the interest-vector of user u in time t+1, $\vec{T}_{u,t+1}\vec{T}_{u,t+1}$. Then we can get recommending score of article i for user u in time t+1 with following formula:

Figure 4. Visiting log of user "9106148"

user_id	news_id	visit_time	news_title
9106148	100642596	2015-03-15 16:22:03	【刘汉兴衰调查之四】周滨的生意伙伴
9106148	100651913	2015-03-15 16:24:03	航班去向不明警方开始调查机组人员
9106148	100642302	2015-03-15 16:25:02	【刘汉兴衰调查之一】兄弟红黑路
9106148	100642684	2015-03-15 16:28:55	刘汉与五矿的暗黑交易
9106148	100642309	2015-03-15 16:30:01	【刘汉兴衰调查之二】斗法袁宝璟
9106148	100642311	2015-03-15 16:32:46	【刘汉兴衰调查之三】炒股金路集团内幕
9106148	100642596	2015-03-15 16:34:30	【刘汉兴衰调查之四】周滨的生意伙伴
9106148	100642609	2015-03-15 16:34:56	【刘汉兴衰调查之五】矿业大亨
9106148	100642618	2015-03-15 16:36:22	【刘汉兴衰调查之六】豪赌江湖

Figure 5. Statistics of time difference of reading time and delivery time

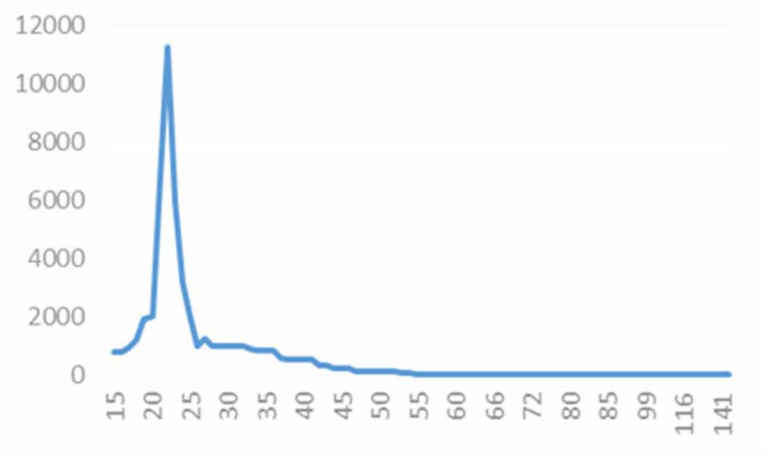
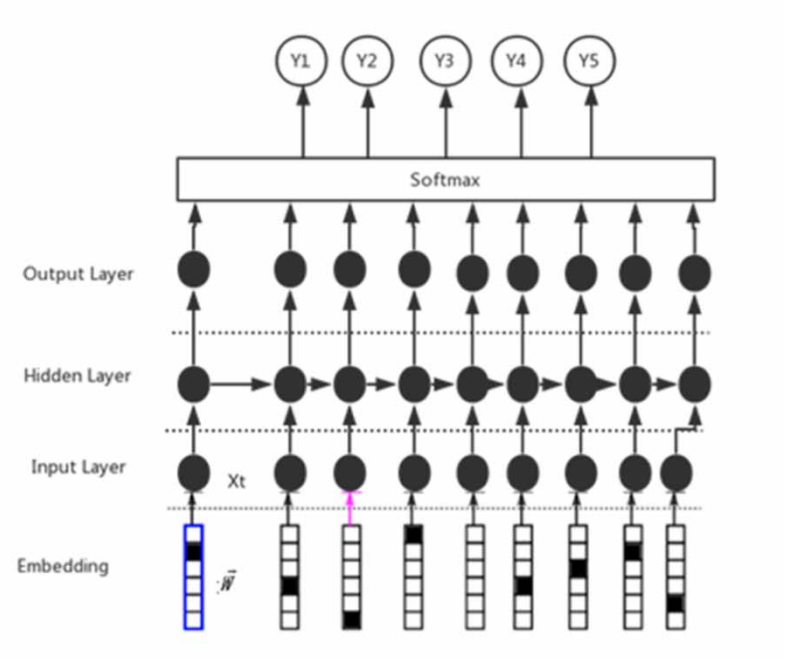


Figure 6. $\overrightarrow{W_i}$ **xt**



$$score_{ui}^{t+1} = \vec{T}_{u,t+1} * \vec{w}_i \tag{1}$$

Initial recommendation results can be calculated with formula:

$$score_{ui}^{0} = \vec{t}_u * \vec{w}_i \tag{2}$$

The picture in Figure 7 presents the architecture of LDA-LSTM recommendation algorithm.

3.2. Training of Algorithm

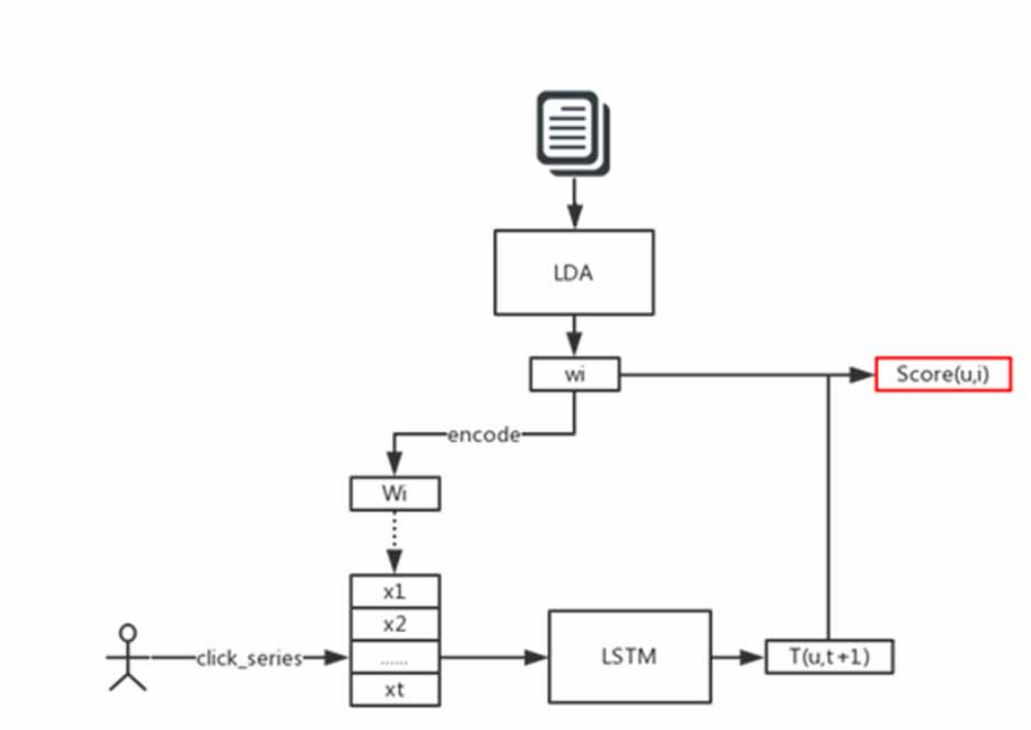
The LDA-LSTM recommendation algorithm is divided into LDA part and LSTM part. The LDA part is trained with the EM algorithm to get the topic vector $\vec{w}_i$ of We-media articles. Then, according to the formula of sigmoid:

$$sigmoid\left(x\right) = \frac{1}{1+e^{-x}} sigmoid\left(x\right) = \frac{1}{1+e^{-x}} \tag{3}$$

we can get the users' long-term historical interest vector $\vec{t}_u$.

The input data for the training of LSTM part is the topic-coding sequence $\vec{W}_k \vec{W}_k$. The target function of LSTM part is:

Figure 7. Architecture of LDA-LSTM recommendation algorithm



$$L\left(x,y\right)=-\sum_{\vec{w}_k\in click\_set}\log(\vec{T}_{u,t+1}*\vec{w}_k)L\left(x,y\right)=-\sum_{\vec{w}_k\in click\_set}\log(\vec{T}_{u,t+1}*\vec{w}_k) \quad (4)$$

3.3. Sequential Alignment and Parallelization Design

A separate training each time is required by traditional recurrent neural networks (or its variants like LSTM) when the lengths of training sequences are not equal. It is not feasible for hundreds of millions of training data. What we need to do is get the data sequences filled. Taking the longest sequence length as the number of rows of the matrix, the rest of the sample data to fill with zeroes. Then the batch size training the batch size of samples together.

The chart in Figure 8 presents a specific example to describe the process. From the extract of We-media article training samples, in order to facilitate understanding, we only choose a batch size of 5. Sample1 is the longest click sequence with 8 items. So the other samples are filled with 0 to make each length of each sequence to 8.

With the samples above, the chart in Figure 9 presents the corresponding masking-code.

Design of "fill 0" operation can solve the problem that different users' sequences are not the same length. Let the manipulate data of multiple users do matrix operations simultaneously. But another problem is, at the time of each LSTM layer training, state transition output and hidden layer nodes output calculation is overall consideration and the current zero padding to the training samples, in theory, should not be updated. With the ideas of dynamic programming, we can solve this problem. Each time when we get the current value of the cell state and hidden layer, we can calculate the value of current state and the previous state as a linear combination. In particular, if in time t, clicking status are currently being updated as below.

Figure 8. Sequences are filled with 0

	T1	T2	T3	T4	T5	T6	T7	T8
Sample1	10110	11010	00011	11010	01010	01110	01011	10110
Sample2	01011	10010	10010	10010	10010	00000	00000	00000
Sample3	11010	11000	01010	00011	0110	01011	11010	00000
Sample4	01110	00110	11010	01011	10010	00000	00000	00000
Sample5	00111	10010	00000	00000	00000	00000	00000	00000

Figure 9. Masking-code

	T1	T2	T3	T4	T5	T6	T7	T8
Sample1	1	1	1	1	1	1	1	1
Sample2	1	1	1	1	1	0	0	0
Sample3	1	1	1	1	1	1	1	0
Sample4	1	1	1	1	1	0	0	0
Sample5	1	1	0	0	0	0	0	0

Input Gate:

$$i_t = sigmoid\left(W_i * h_{t-1} + U_i * x_t + b_i\right) i_t = sigmoid\left(W_i * h_{t-1} + U_i * x_t + b_i\right) \tag{5}$$

Forget Gate:

$$f_t = sigmoid\left(W_f * h_{t-1} + U_f * x_t + b_f\right) f_t = sigmoid\left(W_f * h_{t-1} + U_f * x_t + b_f\right) \tag{6}$$

Output Gate:

$$o_t = sigmoid\left(W_o * h_{t-1} + U_o * x_t + b_o\right) o_t = sigmoid\left(W_o * h_{t-1} + U_o * x_t + b_o\right) \tag{7}$$

Cell state:

$$\tilde{c}_t = \tan h(W_c * h_{t-1} + U_c * x_t + b_c) \tilde{c}_t = \tan h(W_c * h_{t-1} + U_c * x_t + b_c) \tag{8}$$

Traditionally we update the migration status with the formula blow:

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{9}$$

We also need to perform the following combination operations at the end:

$$c_t = mask * c_t + \left(1 - mask\right) * c_{t-1} c_t = mask * c_t + \left(1 - mask\right) * c_{t-1} \tag{10}$$

The formula is made up of two parts. If the masking-code has a value of 1, the left part of the plus sign updates the current state. If the mask is 0, then the state is returned to the last non-zero state. The chart in Figure 10 shows that when running to T7, the new data is updated directly for sample1 and sample3. But for sample2, sample4, and sample5, the backtrace is performed.

Figure 10. Backtrace of zero state

	T1	T2	T3	T4	T5	T6	T7	T8
Sample1	10110	11010	00011	11010	01010	01110	01011	10110
Sample2	01011	10010	10010	10010	10010 ←	00000 ←	00000 ←	00000
Sample3	11010	11000	01010	00011	0110	01011	11010 ←	00000
Sample4	01110	00110	11010	01011	10010 ←	00000 ←	00000 ←	00000
Sample5	00111	10010	00000 ←	00000 ←	00000 ←	00000 ←	00000 ←	00000

We update the value of hidden layer also with the same idea, except for the following regular updates:

$$h_t = o_t * \tanh(c_t) \tag{11}$$

In the end, the following additional combination operation is required:

$$h_t = mask * h_t + \left(1 - mask\right) * h_{t-1} h_t = mask * h_t + \left(1 - mask\right) * h_{t-1} \tag{12}$$

3.4. Dropout of LSTM Model

LSTM is deep neural net with a large number of parameters, overfitting is a serious problem of LDA-LSTM algorithm. Dropout (Srivastava N, 2014) is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data. What its working principle is, during each iteration, that dropping out some of net units randomly, so the corresponding weights will not be updated this time. The picture in Figure 11 presents the updated.

After adding Dropout method, the picture in Figure 12 presents the training of LDA- LSTM algorithm.

4. IMPLEMENTATION OF RECOMMENDATION SYSTEM

LDA - LSTM recommendation algorithm is combined with LDA part and LSTM part. The LDA part is mainly responsible for the extraction of topic vector of We-media articles. The requirement of timeliness of We-media content processing is high. WeChat official accounts produce a large number of articles every day which is a challenge of the LDA part.

To meet the requirements above. We can make implementation of this part on the Spark (Karau H, 2015) distributed cluster offline. The results of LDA part can be stored into HDFS or HBase. The picture in Figure 13 shows the structure of LDA part.

The LSTM part is mainly responsible for updating recommendations based on the user's short-term session behavior. But the training of the LSTM model is time-consuming. So, we train LSTM model offline. The picture in Figure 14 presents LSTM training process of the model. Firstly, we will collect user historical topic-coding sequences and sent them to a fixed server node with Tensorflow

Figure 11. Dropout method

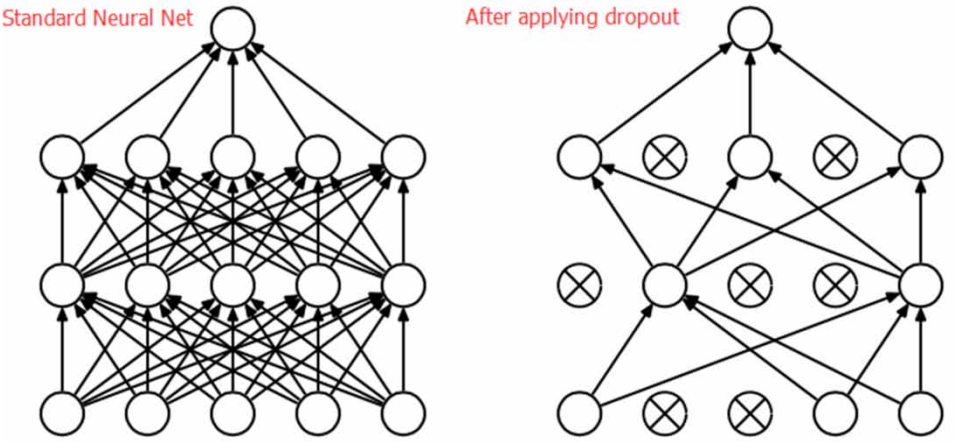
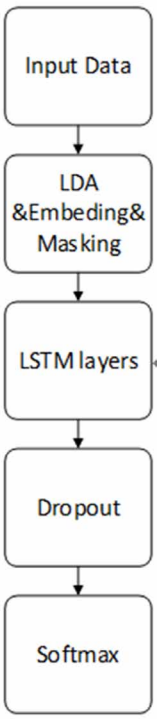


Figure 12. Training of LDA-LSTM algorithm



(Karau H, 2015). Secondly the fixed node trains LSTM model with Tensorflow framework and dump LSTM model to file. Finally, the dump file will be uploaded to Spark file system. The main information that stored in the model file is the parameters of each gate on the LSTM block.

After LSTM model was trained and dump to file, the online recommendation part will load the model file. Users generate reading sequences through clients. Spark Streaming will handle

Figure 13. The structure of LDA part

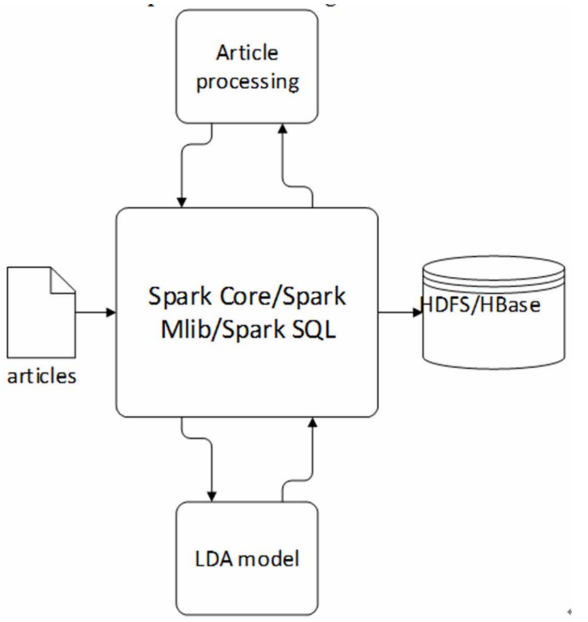
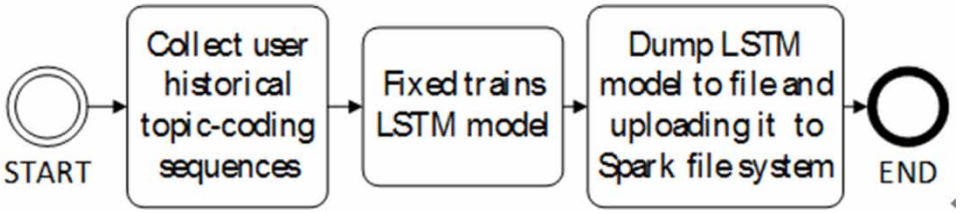


Figure 14. Offline training of LSTM model



those reading sequences to the topic-coding sequences. The LSTM model is used to predict interest distribution of users. The picture in Figure 15 presents the working diagram.

5. EXPERIMENTS

We collected 5866 we-media articles and 10000 users' visiting log during March 1, 2016 to March 31, 2016. I divided those data into two sets. Set of March 1 to March 30 is used to train model and set of March 31 is used to test model.

In order to compare the performance of different algorithms, I also tested some other algorithms such as random recommendations, Most-popular recommendation and Collaborative filtering (CF). The final result is shown in the Table 1.

Thus, the LDA-LSTM algorithm has better performance on we-media article recommendation.

To observe the performance of solving a "New Item Problem" which is an important characteristic of We-media user, we filter recommending results of March 30 from model above. The testing result is shown in Table 2.

Figure 15. Online recommendation

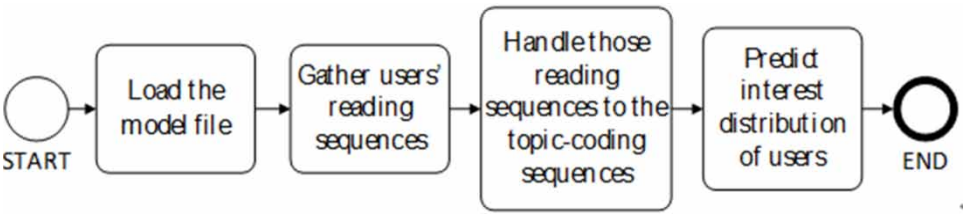


Table 1. Testing result

Algorithm	Precision	Recall	F1
LDA-LSTM	20.01%	2.10%	3.8%
User-CF	18.18%	1.81%	3.29%
Most-Popular	9.10%	0.91%	1.65%
Random	1.18%	0.32%	0.5%

According to the results of Table 2, the LDA-LSTM recommendation algorithm has the best performs on the new articles. The "New Item Problem" is solved by using of information of article topics.

Table 2. Testing result of "New Item Problem"

Algorithm	Precision	Recall	F1
LDA-LSTM	3.65% 3.65%	1.99%	3.8%
User-CF	3.17%	1.82%	3.29%
Most-Popular	2.22%	0.87%	1.65%
Random	1.17%	0.30%	0.5%

6. CONCLUSION AND FUTURE WORK

LDA-LSTM algorithm initialize recommendation with user's interest-vector $\vec{t}_u$. That is idea of content-based recommendation which makes results of recommendation more personalized than that of Most-popular algorithm. Considering we-media users' reading habits of fragmentation reading, speed reading and serialization reading, LDA-LSTM algorithm predicts users' interest vector sequences timely. Experiments on the real datasets show that the combined method outperforms the traditional collaborative filtering recommendation and non-personalized recommendation method.

The implementation of LDA-LSTM algorithm is based on the Spark framework. We divide implementation of LDA-LSTM algorithm in 2 part: LDA part and LSTM part. LDA part can handle hug number of articles with a high time timeliness. LSTM part split the training of LSTM model to offline. But using of LSTM model timely on Spark Streaming.

However, the LDA algorithm works on word-bag method without considering the relationship of words sequences and context semantic (Sundermeyer M, 2014). Thus, the LDA-LSTM algorithm has poor performance on short texts such as WeiBo. In the future study, we can use RNNs model to

extract topics of short texts and get a better effect. At the same time, LDA-LSTM algorithm is only a simple application of LSTM, we can improve it with attention model (Werbos P J., 1990) which is more similar to user's reading behavior. The implementation of LDA-LSTM system's LSTM part is trained on a fixed node which will be the bottleneck of the whole system. In the future, we can use technical solutions like GPU clusters.

ACKNOWLEDGMENT

Dr. Lidong Zhai, affiliated with Institute of Computing Technology of Chinese Academy of Sciences, China and guest editor of the Special issue of Big Data and Computing Security aided with the completion of this project.

REFERENCES

方滨兴, 贾焰, 李爱平, 等. (2016). 大数据隐私保护技术综述. 大数据, (1), 1-18.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). *Neural machine translation by jointly learning to align and translate*. arXiv preprint arXiv:1409.0473

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*(Jan), 993–1022.

Karau, H., Konwinski, A., & Wendell, P. (2015). *Learning spark: lightning-fast big data analysis*. O'Reilly Media, Inc.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. doi:10.1038/nature14539 PMID:26017442

Liang. (2016). Recommender system in action. *Posts & Telecom Press.*, *6*, 1–91.

Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, *7*(1), 76–80. doi:10.1109/MIC.2003.1167344

McLachlan, G., & Krishnan, T. (2017). *The EM algorithm and extensions*. John Wiley & Sons.

Medsker, L. R., & Jain, L. C. (2001). *Recurrent neural networks*. Design and Applications.

Pazzani, M. J., & Billsus, D. (2007). *Content-based recommendation systems. In The adaptive web* (pp. 325–341). Springer.

Penguin Intelligence. (2015). *The first data report of WeChat official accounts*. Retrieved from http://tech.qq.com/a/20150127/018482.htm#p=1

Pham, V., Bluche, T., & Kermorvant, C. (2014). Dropout improves recurrent neural networks for handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on.* IEEE.

Rush, A. M., Chopra, S., & Weston, J. (2015). *A neural attention model for abstractive sentence summarizatio*. arXiv preprint arXiv:1509.00685

Srivastava, N., Hinton, G. E., & Krizhevsky, A. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sullare, V. A., & Thakur, R. S. (2016). Rule Based Recommendation System for Performance Improvement in Engineering Institutions. *Data Mining and Knowledge Engineering*, *8*(1), 11–14.

Sun, F. (2017). *Data source of we-media articles and users' visiting log*. Retrieved from https://github.com/feiqiangs/-data-source-of-we-media-articles-and-users-visiting-log

Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM Neural Networks for Language Modeling. *Interspeech*, 194-197.

Wang, S., & Jiang, J. (2015). *Learning natural language inference with LSTM*. arXiv preprint arXiv:1512.08849

Werbos, P. J. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, *78*(10), 1550–1560. doi:10.1109/5.58337