Deep Learning-based Short Video Recommendation and Prefetching for Mobile Commuting Users

Qian Li
School of Information and Communication Engineering
Communication University of China
liqian0716@cuc.edu.cn

Yuan Zhang
Key Laboratory of Media Audio & Video
Communication University of China
yuanzhang@cuc.edu.cn

Hong Huang
Huazhong University of Science and Technology
honghuang@hust.edu.cn

Jinyao Yan
Key Laboratory of Media Audio & Video
Communication University of China
jyan@cuc.edu.cn

ABSTRACT

Mobile short video application is growing rapidly and it is quickly occupying people's life. In this paper, we consider an emerging yet common scenario of short video application usage: mobile users watching short videos on their daily commuting trip on high speed public transport, where the network condition is unsatisfactory. To reduce users waiting time and improve the QoE, we propose a deep learning-based data recommendation and prefetching scheme which obtains user interests and pushes the preferred short video content to the most likely base station that users will be connected to. We use Principal Component Analysis (PCA) plus dropout to reduce the feature dimensions of Inception structure to improve the short video recommendation speed without degrading the accuracy. Through experimental evaluations, we show that the proposed scheme can effectively recommend short video and predict user trajectory, with a recall rate of 100%.

CCS CONCEPTS

• Networks → Network mobility; Mobile networks;

KEYWORDS

Short video, Recommendation, Prefetching, Mobility Prediction

ACM Reference format:

Qian Li, Yuan Zhang, Hong Huang and Jinyao Yan. 2019. Deep Learning-based Short Video Recommendation and Prefetching for Mobile Commuting Users. In *Proceedings of ACM SIGCOMM 2019 Workshop on Networking for Emerging Applications and Technologies (NEAT 2019). ACM, Beijing, BJ, China, 7 pages.* https://doi.org/10.1145/3341558.3342205

1 INTRODUCTION

Over the last few years, with the coverage of 4G/WiFi and the continuous development of mobile Internet technology, mobile short video Applications such as Douyin and Kuaishou quickly occupies people's work and life. According to [1], the number of short video users is already 648 million in China by the end of 2018, and the number of daily active users is growing at a speed of over 800% each year. Unlike the traditional long video viewing experience where users can download the videos in advance and watch them in an offline way, short video is could only be viewed in an online fashion. This watching paradigm of short videos is proposing unique challenges for short video applications [2,3].

An emerging trend is that people are addicted to watching short videos on public transports during their way to work. In a survey [28] with 190 respondents we find that over 46.47% reported having the experience of watching short videos on their daily commuting time by public transport service. On one hand, since the public transport is usually moving at a high speed, the network condition is usually not stable. According to [25], the Internet Service Providers (ISPs) have complex and diverse handoff policies, which will significantly affect the performance of highly mobility users in cellular network. This is also coherent to our survey that over 71.43% respondents reported having disconnection experience during their watching short videos on public transport. On the other hand, the route of the commuting trip is quite predictable, which provides us the opportunity of reducing the users' waiting by prefetching the users' interested short videos to the next base station in advance. However, the prefetching paradigm would cause waste of network resource and degrade user QoE if the prefetched content is satisfactory. As a result, we believe that a recommendation and prefetching system with high prediction accuracy could become the new growth point for short video Applications.

NEAT' 19, August 19, 2019, Beijing, China

ACM ISBN 978-1-4503-6876-6/19/08…$15.00
https://doi.org/10.1145/3341558.3342205

Based on the above idea, we propose a short video recommendation and prefetching scheme for mobile commuting users. As for the short video recommendation scheme, since the resolution ratio of short video application is small, which indicates less details and simpler texture of the video frame, we propose designing a recommendation algorithm with less features and redundancy. As a result, we propose to use Inception [21] network together with PCA and softmax to reduce the feature dimensions and adjust overfitting problem. One challenge is that as a third party, we do not have the actual viewing trace. As a result we use the user marked liked videos as the positive samples for each user in the training set. In our crawled dataset, the number of unmarked videos dominates the video set. Fig. 1 shows the CDF of the marked positive samples of users. As can be seen from Fig.1, the number of marked positive samples varies tremendously across the crawled 233 Douyin users, from 1 to 3940. Since the total amount of short videos of the 78170, if we use the unmarked videos directly as the negative samples, even the user with maximum marked likes is extremely imbalanced let alone the other users. To address the imbalanced sample problem and maintain the positive to negative ratio as 1:3 (the empirical positive to negative ratio used for video recommendation system), we propose a classification-based balancing scheme using YOLO3. As for the prefetching scheme, we propose to use Long Short-Term Memory (LSTM) to predict the users' mobility pattern and adjust the model for higher prediction accuracy on various aspects such as LSTM parameters and input sequences. Comparison with Recurrent Neural Network (RNN) and Gated Recurrent Unit (GRU) shows LSTM produces the highest prediction accuracy.

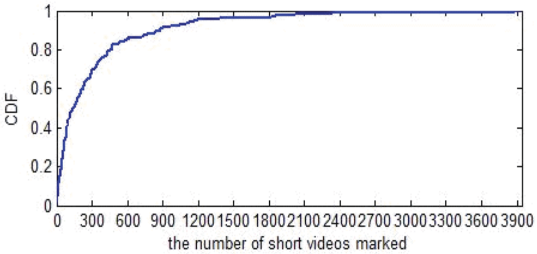

Figure 1: CDF of the number of marked positive samples

In summary, we propose deep learning-based short video recommendation and prefetching system with user mobility prediction for commuting scenario, so as to improve the overall user QoE of less waiting time and better recommendation accuracy. The main contributions are summarized as follows.

- We identify the problems of watching short video in current transmission paradigm of commuting mobile users. To address these problems, we propose a new deep learning-based framework that prefetches recommended short videos for mobile users with user mobility prediction.
- Based on crawled Douyin [29] data set, we use an improved version of GoogLeNet Inception to recommend short videos for each individual user which has a less computational time and yet high recommendation accuracy.
- We use LSTM to predict user mobility patterns using real trace from one of the biggest ISPs in China. We observe the prediction result of the users and regard the predictable users as commuting users. We further explain the prediction results given the users' career information of the ISP dataset.
- Experimental result shows that the recall rate of our recommendation algorithm is 1, which means that all the samples that marked like are recommended for users in the testing set.

The rest of the paper is organized as follows. In Section 2, we give the literature review of related work. Section 3 presents the problem definition and system overview. The design details of the recommendation and prefetching algorithm are described in Section 4. Section 5 presents a thorough evaluation results of the proposed scheme from various aspects. We conclude our work and discuss future directions in Section 6.

2 RELATED WORK

We summarize the related work in terms of user recommendation algorithms and trajectory prediction algorithms.

Recommendation algorithm: the recommendation schemes can be roughly classified into two types: traditional recommendation methods and deep learning methods. Traditional recommendation approaches such as collaborative filtering, content-based recommendation algorithm and hybrid recommendation algorithm have proven fast and effective performance for many recommendation tasks [4-7]. On the other hand, deep learning methods convolutional neural networks (CNN) [16], RNN [14] and LSTM [15] have proven to be successful for more sophisticated recommendation tasks [8-13]. They use multi-source heterogeneous data as input and apply an end-to-end model to automatically train prediction model, thus alleviating the data sparseness and cold start problems faced by traditional recommendation system, and improving the performance of recommendation system. Inspired by this advantage, we design our short video recommendation model by deep learning methods. We propose this recommendation model to learn user preference and improve the accuracy of caching.

Mobility prediction: the mobility prediction problem usually entails a discrete time system therefore the mobility prediction algorithm should be established taking inputs on time sequence. Traditional machine learning approaches such as exponential smoothing and Markov-based [17] algorithms are widely used in prediction algorithms, due to their efficiency, simplicity, and low computing costs. However, their limitation is also distinct, for example Markov-based prediction can only retrieve the short term correlations of sequences and the exponential smoothing approach only works for stable inputs. Deep learning methods such as RNN and their variants including LSTM networks and GRU [18] have proven to be successful for sequence prediction tasks [19, 20]. Among them, LSTM shows a good potential to capture the transition regularities of human movements since they have memory to learn the temporal dependence between observations. Therefore, we use LSTM as our mobility prediction algorithm to predict next collected base station.

3 SYSTEM OVERVIEW

We are considering the short video recommendation and prefetching scheme for commuting users. The overall system architecture is illustrated in Fig. 2, which contains a short video recommendation module and a mobility prediction module. For each user, we use the recommendation algorithm to get her top-5 preferred short videos using an improved version of Inception. After that we use LSTM to predict the next cell ID and push the recommended short videos onto that base station. Since we do not have the commuting user trace, we predict trajectories for all the mobile users and regard the trace with high prediction accuracy as commuting trace.

The two-stage recommendation and prefetching approach allows us to make recommendations from a very large set of short videos while still being certain that the top-5 short videos appearing on the device are personalized and engaging for the user, and cache the top-5 short video in next base station in advance, therefore improving the user QoE in terms of recommendation accuracy and transmission delay.

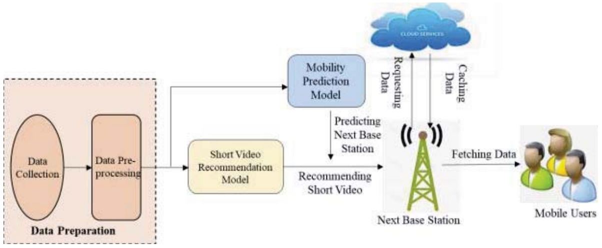

Figure 2: Overall system architecture

4 SYSTEM DESIGN

In this section, we describe the design details of our two-stage recommendation and prefetching scheme respectively.

4.1 Short Video Recommendation

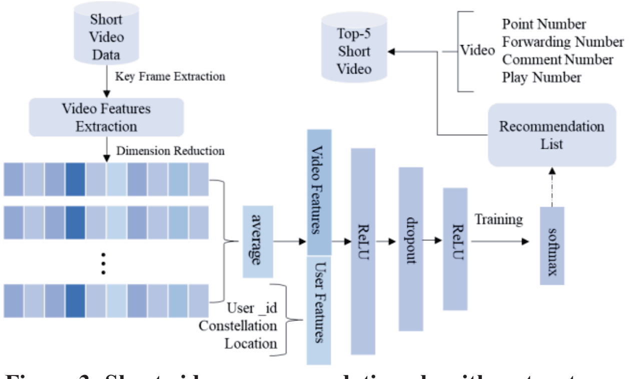

Figure 3: Short video recommendation algorithm structure

We process the videos and extract features using GoogLeNet Inception network [21] like [22]. Instead of obtaining better training effect by increasing the depth of the network, which may cause problems such as overfitting, gradient disappearance, gradient explosion and so on, GoogLeNet improves the training results by using both deeper and broader structure. We use the Inception model of Google on the ImageNet [23], which can theoretically reduce the number of larger parameters and is simple to operate in practical application. To be specific, we extract one key frame per second for each short video using OpenCV (here the key frame is I frame in OpenCV’s implementation), feed all the key frames into Inception to get the feature vector. The feature vector of original Inception structure is 2048 dimensions per frame. Since the short video clips usually use a resolution rate of 540*960, which indicates less details and simpler textures, we can apply PCA to reduce feature dimensions and take an average of the features over all the frame for each video. These two compression techniques greatly reduce the computation time. We further eliminate the possible overfitting problem by adding a dropout layer. The reduced short video features together with user information such as user ID, user location and user constellation etc. are fed into the ReLu activation function of the last hidden layer. The result is cascading to the softmax classification layer, which produces the recommendation list. We rank the selected list of short videos with the video related information such as the number of likes, forwardings, comments and playbacks, and select the top-5 short videos on the recommendation list for each user. The proposed short video recommendation scheme is shown in Fig. 3.

4.2 Mobility Prediction

Since we do not have the pure user commuting data, we use user mobility trace to predict the user mobility pattern and regard the predictable traces as commuting trip. Through data visualization analysis, it is found that the trajectory of users on working days is more regular, connected base stations of user are also limited, which is rational and is consistent with our setting of commuting scenario. Fig. 4 illustrates the mobility prediction module with data preprocessing and LSTM prediction. The mobility prediction means to predict the next cell ID given a sequence of previously connected cell IDs. We use one-hot to encode the cell IDs. Afterwards, the pre-processed data are fed into the LSTM network for training like Fig.5. LSTM adjusts by calculating loss function and accuracy. If the loss function is under a given threshold, and the accuracy is more than another given threshold, the prediction results are output directly, otherwise, the parameters will be adjusted to meet the conditions. The details of training process and parameter adjustments are described in Section 5.3.

5 EXPERIMENTS AND EVALUATION RESULTS

We implement our two-stage scheme using Tensorflow [24]. The experimental result is run on an Intel Core i7 Hexa-Core processor with 16GB RAM and GTX 1080 Ti graphics card.

5.1 Dataset and Preprocessing

User mobility trace: the real-time user mobility trace contains 5,000 users event-driven trace (a log is generated when data transmission happens) from one of the biggest ISPs in China for one week, from November 15, 2015 to November 21, 2015. It includes user ID, access time and cell ID. We apply one-hot to project the cell ID. Since we are considering the commuting scenario, we use data from Monday to Friday for experiment and divide the data into training set and testing set.

Short video dataset: the short video dataset is crawled from Douyin of 78170 records by 233 users. It includes fields like user ID, user avatar, user nickname, user location, user constellation, video ID, video release time, video image, video description, number of likes\comments\forwardings\shares for each video and video file itself.

Sample balancing: to address the imbalanced sample problem, we first use YOLO3 to classify all the crawled short videos into 80 different types using the trained COCO [26] coefficients and sort all the short videos within each type according to their like and forwarding number. Then we obtain the likeness of all the 80 types of each user according to her marked like number. We randomly extract a certain proportion of short video from unmarked video in the least liked types as negative examples. We use 70% of the dataset for training, and the rest 30% for testing.

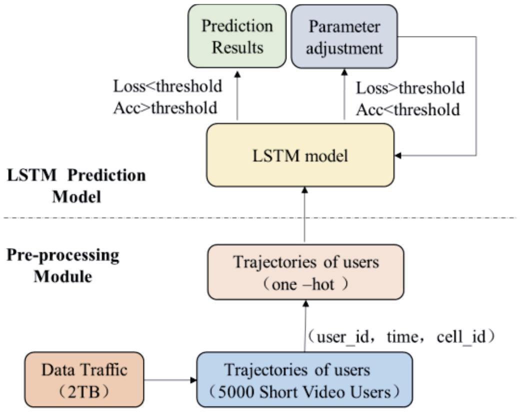

Figure 4: Mobility prediction structure

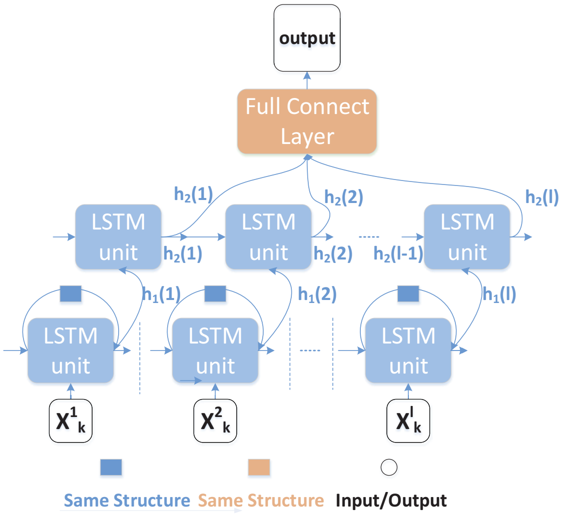

Figure 5: LSTM model

5.2 Short Video Recommendation

Table 1 shows the performance of the recommendation scheme using different loss functions number of nodes in the recommendation network. When the loss function is mean square error the number of nodes is 30, the recommendation accuracy and F1 are the highest, the accuracy reaches 69%, F1 is 0.817. We further adjust the model by varying network depth. As shown in Table 2, when the network depth increases, the performance first improves and then drops. The performance is best when the depth is 5. However, with the increase of network depth, the computation will also increase. The accuracy of 3 is similar to 5, so we set the network depth to 3. The recall rate of our recommendation scheme is 100% using parameter setting of 30 nodes and 3 layers.

Table 1: Performance of short video recommendation scheme using different loss function and number of nodes

		Acc	F1
Loss function	Mean absolute error	0.664	0.798
	Cross entropy	0.667	0.808
	Mean square error	0.688	0.815
Nodes	10	0.67	0.808
	30	0.69	0.817
	50	0.664	0.798
	100	0.664	0.798
	128	0.68	0.814
	160	0.65	0.791
	256	0.66	0.796

Table 2: Recommendation results using different the neural networks layers

Layers	Acc	F1
3	0.69	0.817
4	0.698	0.822
5	0.704	0.826
6	0.641	0.77
7	0.61	0.75
8	0.67	0.8

We also compare our recommendation scheme with objection detection by YOLO. To use YOLO3 for short video recommendation, we still first extract key frames for each short videos. We use video clustering algorithm to cluster the short videos according to the categories obtained by the objection detection, and calculate the video categories that each user likes. The recommendation accuracy for using YOLO with a pre-trained COCO [26] category and pre-trained parameter is only 35%. The reason for the low accuracy is that the objection category of short videos from Douyin are mainly person and cannot be distinguished from each other since COCO classify all persons as a single category regardless of age, gender, occupation, etc. We also train YOLO with the Neural Baby Talk [27] dataset which breaks down the “big” category of person in COCO and provides a rich subdivision of person. Since Neural Baby Talk doesn’t provide pre-trained parameters, we need to train the parameters by ourselves. The recommendation accuracy for using YOLO with Neural Baby Talk category is 52% which performs better than COCO category but still worse than our algorithm.

We further test the effect of PCA and dropout layer. As shown in Fig. 6, our recommendation algorithm is denoted as PCAdrop has the lowest computation time compared to pure PCA and original Inception. The recommendation accuracy shown in Fig. 7 indicates that our PCA with dropout design can achieve comparable recommendation accuracy w.r.t. original Inception and PCA plus Inception when the number of layers is small, moreover it even performs better than the other two candidates when the number of layers grow larger. It probably indicates the PCA plus dropout may have less overfitting problem than pure PCA and original Inception.

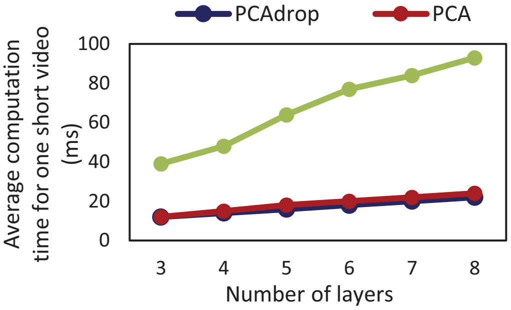

Figure 6: Comparison to original Inception structure w.r.t. computation time

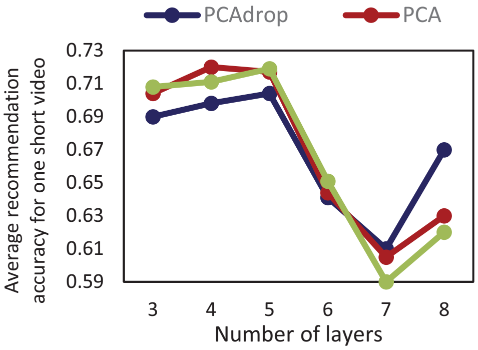

Figure 7: Comparison to original Inception structure w.r.t. recommendation accuracy

5.3 Mobility Prediction

Because trajectory prediction is a sequence-related problem, the accuracy of the prediction results is closely related to the length of the input sequence. Fig. 8 shows the trajectory prediction results of a user, the prediction accuracy changes with the increase of input sequence. As number of input sequence increases, the mobility prediction accuracy increases firstly and then decreases, when the number of input sequence equals 10, the accuracy is the highest. Fig. 9 shows the trajectory prediction results of the same user, the prediction accuracy changes with the increase of training times. With the increase of training times, the accuracy increases to a stable level, when the times of training equals 7, the accuracy is better with the higher calculation efficiency.

We further compare our LSTM prediction model with other algorithms. As can be seen in Fig. 10, for the predictable users the prediction accuracy of LSTM is much higher than RNN and GRU. For users whose prediction accuracy is less than 0.4, we can visualize their mobile trajectory, and find that there is no regularity in their base station connection trajectory in one week.

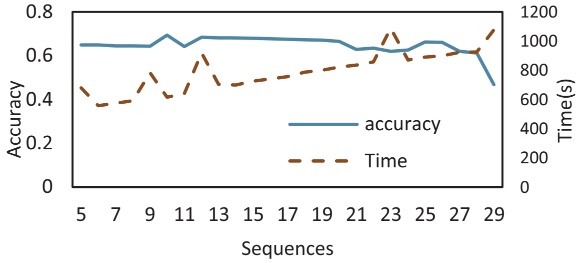

Figure 8: User mobility prediction using different sequences

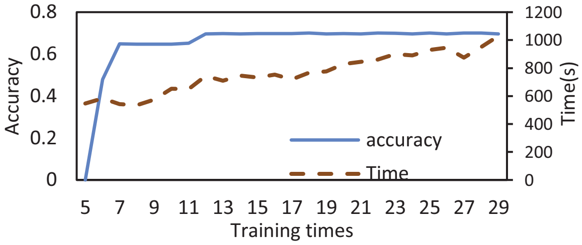

Figure 9: Mobility prediction using different layers

By looking at their occupation information, we can find that these users belong to the public industry, traffic industry or energy industry. Therefore, we have reason to infer that the inaccuracy of these mobile user mobility prediction is caused by their working nature. Another possible reason is that the data period we obtain is not long enough to reflect the periodicity of their trajectory.

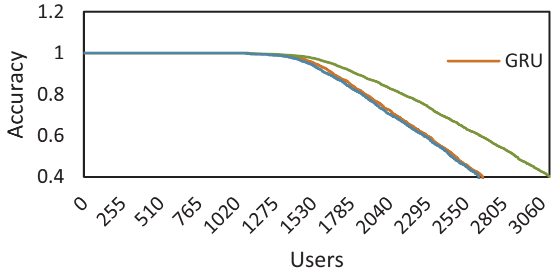

Figure 10: Comparison of user mobility prediction algorithms

5.4 Feasibility Assessment

We test the network condition in commuting case for both subway and bus. In our experiment, the network condition is very unstable, the highest download rate can reach 7.5Mbps, the lowest is only 4.8Mbps, and the average download rate is 5.15 Mbps. The maximum size of the crawled short video files is 5.2MB, and is on average 2.5MB. While the crawled video length is between 7.2s to 25.3s, therefore, in the worst case, the overall recommendation and transmission time of the short videos is $12ms + 5.2 \times 8/4.8\,s = 8.7s$. It means that for the worst case, the user would wait for $8.7 - 7.2 = 1.5s$ to get her preferred short video clips using our prefetching scheme, which is quite tolerant compared to the $8.7s$ waiting time without prefetching.

6 CONCLUSIONS

In this paper, we propose deep learning-based short video recommendation and prefetching for mobile commuting users which obtains user interests and pushes the preferred short video content to the most likely base station that users will be connected to. By experimental evaluation, we show that our two-stage scheme can provide better recommendation result and eliminate waiting time of mobile short video users.

ACKNOWLEDGMENTS

This work is supported by "the Fundamental Research Funds for the Central Universities". National Natural Science Foundation of China (No. 61802140), Hubei Provincial Natural Science Foundation (No. 2018CFB200).

REFERENCES

[1] The 43rd statistical report on the development of Internet in China.2018.http://www.cac.gov.cn/201902/28/c_1124175677.htm

[2] Dong Liu; Binqiang Chen, Chenyang Yang and Andreas F. Molisch. 2016. Caching at the wireless edge: design aspects, challenges, and future directions. IEEE Communications Magazine, vol. 54, no. 9, 22-28, September 2016.

[3] Yuchao Zhang, Pengmiao Li, Zhi-Li Zhang, Bo Bai, Gong Zhang, Wendong Wang and Bo Lian Challenges and Chances for the Emerging Short Video Network. In 2019 INFOCOM WKSHPS.

[4] John S. Breese, David Heckerman, and Carl Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (UAI'98).

[5] Raymond J. Mooney and Loriene Roy. 2000. Content-based book recommending using learning for text categorization. In Proceedings of the fifth ACM conference on Digital libraries (DL '00). ACM, New York, NY, USA, 195-204.

[6] Marko Balabanović and Yoav Shoham. 1997. Fab: content-based, collaborative recommendation. Commun. ACM 40, 3 (March 1997), 66-72.

[7] Przemysław Kazienko and Michał Adamski. 2007. AdROSA-Adaptive personalization of web advertising. Inf. Sci. 177, 11 (June 2007), 2269-2295.

[8] Yuyun Gong and Qi Zhang. 2016. Hashtag recommendation using attention-based convolutional neural network. IJCAI'16.

[9] LI, Yang; LIU, Ting; Jing JIANG; and ZHANG, Liang. Hashtag recommendation with topical attention-based LSTM. Proceedings of the 26th International Conference on Computational Linguistics: Osaka, Japan, 2016, 943-952.

[10] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-Rate Deep Learning for Temporal Recommendation. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '16). ACM, New York, NY, USA, 909-912.

[11] Trapit Bansal, David Belanger, and Andrew McCallum. 2016. Ask the GRU: Multi-task Learning for Deep Text Recommendations. In ACM Proceedings RecSys '16.

[12] Cheng Yang, Maosong Sun, Wayne Xin Zhao, Zhiyuan Liu, and Edward Y. Chang. 2017. A Neural Network Approach to Jointly Modeling Social Networks and Mobile Trajectories. ACM Trans. Inf. Syst. 35, 4, Article 36 (2017), 28 pages.

[13] Paul Covington, Jay Adams and Emre Sargin. Deep Neural Networks for YouTube Recommendations[C] ACM Conference on Recommender Systems. ACM, 2016:191-198.

[14] Hopfield, and J. J. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences 79.8(1982):2554-2558.

[15] HOCHREITER S, SCHMIDHUBER J. Long Short-Term Memory [J]. Neural Computation, 1997, 9(8): 1735-80.

[16] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Comput. 1, 4 (December 1989), 541-551.

[17] Anthony J. Nicholson and Brian D. Noble. 2008. BreadCrumbs: forecasting mobile connectivity. In Proceedings of the 14th ACM international conference on Mobile computing and networking (MobiCom '08). ACM, New York, NY, USA, 46-57.

[18] Junyoung Chung, Caglar Gulcehre KyungHyun Cho, Y.Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. 2014.

[19] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei and Silvio Savarese. 2016. Social LSTM: Human Trajectory Prediction in Crowded Spaces[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 11 pages.

[20] Jun Liu, Amir Shahroudy, Dong Xu, Alex C. Kot and Gang Wang.2016. Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition[J]. 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands.

[21] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision, CVPR 2016.

[22] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, Geoge Toderici, Balakrishnan Varadarajan and Sudheedra Vijayanarasimhan. YouTube-8M: A Large-Scale Video Classification Benchmark, CVPR 2016.

[23] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'15), Francis Bach and David Blei (Eds.), Vol. 37. JMLR.org 448-456.

[24] TensorFlow. 2018. https://www.tensorflow.org/

[25] Haotian Deng, Chunyi Peng, Ans Fida, Jiayi Meng, and Y. Charlie Hu. 2018. Mobility Support in Cellular Networks: A

Measurement Study on Its Configurations and Implications. IMC '18, 147-160

[26] http://mscoco.org/

[27] https://github.com/jiasenlu/NeuralBabyTalk

[28] https://www.wjx.cn/m/37307561.aspx

[29] https://www.douyin.com/