Research on User Preference Film Recommendation Based on Attention Mechanism

Lei Zhu, Yufeng Liu(✉), Wei Zhang, and Kehua Yang

College of Computer Science and Electronic Engineering, Hunan University,
Changsha 410082, China
fx_yfliu@163.com

Abstract. Due to the influence of different factors such as environment, age, and interest, everyone has a different taste and appreciation of the movie. These factors can be used to make more personalized recommendation calculations for movie recommendations, but many traditional methods do not integrate these factors well. Therefore, how to effectively extract key information from multiple directions in a recommendation system is still a challenging problem.

In this paper, we use user, film, and movie scoring data as input, and use the CNN-BLSTM deep learning model that incorporates the multi-head attention mechanism as a training, and finally combine the output features to calculate user preferences for recommendation. The convolutional neural network and LSTM are used to extract the user and movie feature information from the matrix, and the multi-head attention mechanism can also extract the key information from the data. The comparative experimental analysis of different models shows that our proposed user preference model based on attention mechanism can obtain better performance than traditional extraction methods.

Keywords: Multi-head attention · Recommendation system · User preference · Deep learning

1 Introduction

Material life is constantly improving, and watching movies one by one is an entertainment way for people to relax daily. In movie recommendation, the traditional collaborative filtering recommendation algorithm mainly uses users' ratings for movies as the basis for recommendation. And other characteristics of users and movies, such as the user's personal attributes, movie attribute categories, and the relationship between user history and movies. Most of them are not fully utilized, and the disadvantage of this is the lack of fine positioning of user behavior preferences.

The study of deep learning shows great potential in effective learning. Deep learning can be used to automatically learn features, such as learning effective feature representations from text content, and then extracting valid information from the features.

X. Sun et al. (Eds.): ICAIS 2020, LNCS 12240, pp. 428–439, 2020.
https://doi.org/10.1007/978-3-030-57881-7_38

However, many existing models and methods still fail to extract key information from the data.

For deep learning models, such as RNN [1], this type of method does not get rid of the limitation of timing, that is, it cannot be parallelized, which leads to speed and efficiency problems on large data sets. Another example is CNN [2], which is convenient for parallelism and easy to capture some global structural information, but only has the advantage of feature detection but lacks feature understanding.

In order to overcome these difficulties, this paper proposes a user preference learning model that integrates the attention mechanism. Mainly have the following advantages: (1) Use deep learning models to automatically learn deep-seam semantic information without relying on any manually designed feature extraction rules. (2) We introduce a multi-head attention mechanism into the model, which can adaptively combine context information and extract text features from it, thereby improving the accuracy of key information acquisition. (3) In multiple dimensions, try to combine the movie's profile information to better extract features at a deep level.

The rest of this article is organized as follows. The related work is described in Sect. 2. Section 3 details the user preference learning model based on the attention mechanism. Section 4 demonstrates the model through experiments. The detailed conclusions and expectations are summarized in Sect. 5.

2 Related Work

In the current recommendation algorithm research, the collaborative filtering algorithm is one of the most enthusiastic and recommended algorithms by researchers, but the traditional collaborative filtering algorithm will inevitably face cold start and data sparsity [3]. In contrast, deep learning-based recommendation systems lack extensive attention and comment. Deep neural network models are now increasingly used in a variety of tasks, including text summaries, information retrieval, and relationship recognition [4, 5, 32].

In the study of deep learning, Socher et al. [1, 6] used recurrent neural networks to learn the logical relationships between sentences, and performed sentiment analysis and sentence semantic relationship recognition. Kim [2] uses the word vector trained by word2vec to map the sentence sequence into a two-dimensional feature matrix, and uses the convolutional neural network to extract the features of the sentence. Tang et al. [7] realized text-level sentiment classification based on the composition principle and relationship of text semantics, combined with convolutional neural networks and cyclic neural networks. Xie et al. [8] proposed the DKRL model, which uses entity description to represent entity vectors, and uses continuous word bag model and convolutional neural network to encode the semantics of entity descriptions, but this model does not consider the screening of entity information.

With the continuous development of deep learning, the use of attention mechanisms to optimize neural networks [9, 11] has become a hot topic. Bahdanau et al. [10] applied the attention mechanism to the NLP field for the first time, and simultaneously translated and aligned on the machine translation task, and finally achieved good performance. Later, Luong et al. [12] proposed a global attention mechanism and a local attention mechanism, providing a method for calculating the extended attention. Xu et al. [13] proposed the joint representation model of BLSTM, using the attention mechanism to select the relevant information in the entity description, and designing the gate mechanism to control the weight of the structural entity information. This method has a significant improvement in performance, but the hidden state of the BLSTM model needs to be generated in order, resulting in no parallel processing during training, thus reducing efficiency.

Although, deep learning has a good ability to capture features, but different models have different limitations. We propose a user preference learning model that integrates the attention mechanism. It not only considers the screening of the entity information, but also captures the feature information. It is enough to build a long-distance dependency between sentences, to solve the limitation of distance on features, and to have parallelization characteristics. Through experimental verification, our proposed method performs better than the previous method.

3 Model Building

This chapter proposes a user preference learning model that integrates the attention mechanism. The model mainly includes CNN-BLSTM feature extraction module, Multi-head attention module and recommendation table generation module. A schematic diagram of the model is shown in Fig. 1. Firstly, the user's features are extracted by multi-layer neural network. Then the feature matrix is extracted by CNN-BLSTM, and the scoring matrix is reduced in dimension, placed together in the embedding layer. Next, the user features and the movie features processed by the attention mechanism are further processed in the fully connected layer. Finally, the vector obtained by the joint calculation is used to calculate the similarity, thereby implementing the recommended task.

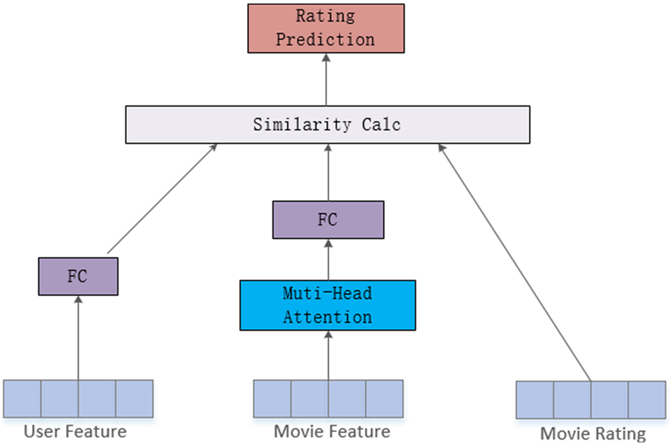

Fig. 1. User preference learning model architecture diagram with attention mechanism.

3.1 Questions Raised

In the aspect of film recommendation, the user's rating of the film as the basis for recommendation [14, 15] is the main processing method of the traditional collaborative filtering recommendation algorithm. A user who has seen the same or similar movie is generated by generating a preference model for the user's history and then filtering through the system. We recommend a movie from other user history to a user who needs to be recommended to generate a recommendation list for that user.

The traditional recommendation algorithm often only uses the scoring data [16], and the user and other various feature data of the movie, such as user age, job or gender, as well as the introduction, type or label of the movie [17] are not able to make full use of it. The result of this is a lack of fine-grained description of user behavior preferences, and in actual recommendations, accurate analysis will greatly improve recommendation accuracy.

Deep learning can not only learn nonlinear multi-level abstract feature representation, but also acquire features that are often dense and low-dimensional [18], which is not available in the traditional collaborative filtering recommendation model. We capture key information by inputting text data into a convolutional neural network and extracting features [19]. At the same time, using the attention mechanism to automatically learn the hidden relationship between the user and the movie features, and combined with the relationship between the film scores, can more effectively reflect the behavior preferences of different users, and thus improve the accuracy of the recommendation system.

3.2 Feature Extraction

The user preference learning model that integrates the attention mechanism not only captures the key information of users and movies, but also considers the interdependence between users and movies, and realizes the division and mining of effective information between users and movies. In order to further capture the relevant information between users and movies, this paper also introduces the retrofit model of attention mechanism, multi-head attention mechanism, and integrates CNN-BLSTM model to solve this problem more effectively.

Feature Extraction Based on CNN-BLSTM. We use the brief information of the movie as a sequence of sentences, and then use the CNN-BLSTM model to divide the brief information of a single movie into multiple sentences. In this way, compared to using an entire statement as input, the feature representation can be extracted sequentially through the convolutional layer neural network, and then these sentence features are sequentially integrated using LSTM [17, 18], and then the entire sentence feature representation is constructed, so that we can get the key information of the text more accurately. Figure 2 is a state diagram of the CNN-BLSTM model.

The first layer is the input layer. We use the input sequence of the movie represented by matrix $M_i^d = [m_1 \ldots m_d]$. The matrix $U_i^d = [u_1 \ldots u_d]$ represents the user's input matrix, and then the movie input sequence is used as the input of the model. The user's input matrix can be directly assigned to the convolutional neural network to extract features.

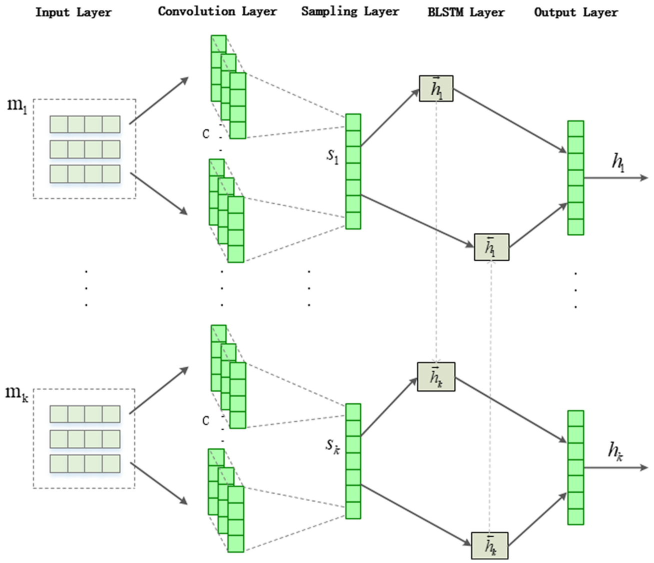

Fig. 2. CNN-BLSTM model diagram.

The second layer is the convolution layer. Drawing on the structure of the CNN model proposed by Kim [2, 19, 31], multiple convolution filters are used to extract multiple sets of local feature maps in the convolutional layer. Given the input matrix S_{ij}, for each row vector m in the matrix, a convolution operation is performed using a filter with a window size of k. The result of the convolution mapping can be expressed as:

$$y_i = \mathcal{F}(W_c \otimes \mathrm{m}_{i:i+k-1} + b_c) \tag{1}$$

Where y_i is the *ith* element of the feature map, W_c is the coefficient matrix, b_c is the bias term, and $\mathrm{m}_{i:i+k-1}$ is the local word window composed of k words. When the word window is stepped from $\mathrm{m}_{i:k}$ to $\mathrm{m}_{n-k+1:n}$, and we can get a feature map:

$$\mathrm{c} = \{y_1, \ldots, y_{n+k-1}\} \tag{2}$$

The third layer is the sampling layer. The most representative features in each feature map are extracted at the sampling layer to obtain a feature representation at the sentence level. At the sampling layer, the feature map is sampled by the max-over-time pooling method proposed by Gollobert [20], and obtaining the feature values $\hat{\mathrm{c}} = \max\{c\}$. The convolutional layer makes one of the filter structures that extract a local feature by the size of the window. We use the structure of h kinds of filters, and take into account the information between contexts by extracting m feature maps in each filter. A feature map obtained by extracting features from all types of filters, after the maximum pooling operation of the sampling layer, obtain the feature value s of length h × m as the output:

$$\mathrm{s} = \left\{\hat{\mathrm{c}}_{1,h_1},\ \ldots,\ \hat{\mathrm{c}}_{m,h_1},\ \ldots,\ \hat{\mathrm{c}}_{1,h_k},\ \ldots,\ \hat{\mathrm{c}}_{l,h_k},\ \ldots\right\} \tag{3}$$

Where c is the *lth* feature value $(1 \leq l \leq m)$ produced by the filter of the *kth* type $(1 \leq k \leq h)$.

The fourth layer is the LSTM layer. The main purpose of this layer is to control the historical information retained by each LSTM unit and to memorize the currently entered information [18, 21, 22, 30], retain important features and discard unimportant features. This layer mainly contains input gates, forgetting gates, cell states and output gates.

$$i_t \ = \ \sigma(W_i s_t + W_i h_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma\big(W_f s_t + W_f s_{t-1} + b_f\big) \tag{5}$$

$$g_t = \tanh\big(W_g s_t + W_g s_{t-1} + b_g\big) \tag{6}$$

$$o_t = \sigma(W_o s_t + W_o s_{t-1} + b_o) \tag{7}$$

Contains the previous cell state and based on current input and last hidden state information generated new information $c_t = i_t g_t + f_t c_{t-1}$ (The initial memory unit c_0 is marked as 0). Finally, the current hidden state of the output is obtained by multiplying the current cell state by the weight matrix of the outputs:

$$h_t = o_t \tanh(c_t) \tag{8}$$

Therefore, for a long sentence consisting of K clause sequences, the sentence feature vector $\mathrm{h}_1, h_2, \ldots, h_K$ is obtained by CNN and bidirectional LSTM.

Multi-head Attention. At present, with the deepening of deep learning research, the attention mechanism has been widely applied to various tasks of natural language processing based on deep learning [23]. The method is based on the following assumptions: the scores between different user relationships and the degree of relevance between movies are different; for important relationships or ratings, more attention is assigned, while others are less focused [24, 25]. The key is how to independently distribute attention without accepting other information [26].

Given a *Query* in the *Target*, by calculating the correlation between it and the *key* of each data pair in the *Sourse*, the weight coefficients of the Key and Value are obtained. We only need to weight the weights separately to the *value*. Can finally calculate the value of Attention.

$$\mathrm{Attention}(Query, Source) = \sum\nolimits_{i=1}^{l} Similarity(Query, Key_i) \cdot Value_i \tag{9}$$

In order to better learn global dependence information from internal text, we used the multi-head attention mechanism proposed by Vaswani et al. [27]. The multi-head attention mechanism takes parallel calculations of multiple scaled dot products, rather than just one calculation. Then, the independent attention calculation units are spliced together, and finally converted into a dimensional output of a desired size by a linear unit.

As shown in Fig. 3. We do n linear mapping of Q, K and V matrices and learn different linear projection matrices $d_n \times d_q$, $d_n \times d_k$ and $d_n \times d_v$. Then the resulting projection

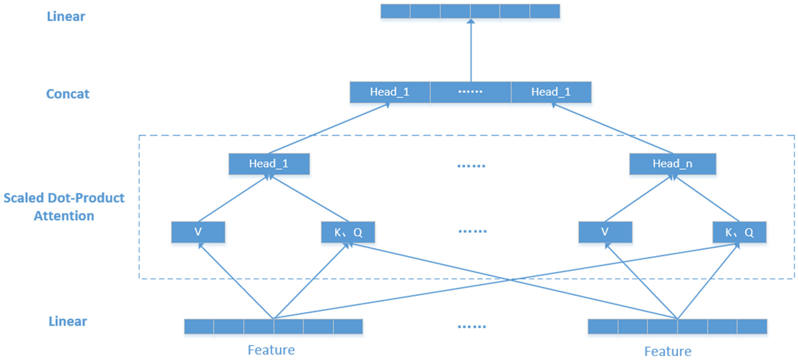

Fig. 3. Multi-head attention model.

matrix will be executed separately "Scaled Dot-Product Attention" The self-attention mechanism is essentially $X = Query = Key = Value$, which means that the Attention is used to find the interdependence within the sequence inside the sequence. The internal self-attention mechanism can solve the problem of weakening the dependence caused by the text being too long. Finally, the matrix of the output $d_n \times d_k$, and connect these values and project again, we can get the final value.

We map Q, K, and V to subspaces of lower dimensions and then perform Attention calculations in different subspaces. A lower subspace dimension reduces the amount of computation, which facilitates parallelization and captures features that represent different subspaces at different locations. This is compared to directly using the linearly mapped Q and K dot product as weighting coefficients, and then weighting and summing V. The advantage is that there will not be a situation where the dot product is too large, and there will be no problem that the gradient is too small. We scale the dot product by $\frac{1}{\sqrt{d_k}}$ to get:

$$head_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) = softmax\left(\frac{QW_i^Q\left(KW_i^K\right)^T}{\sqrt{d_k}}\right)VW_i^V \tag{10}$$

Finally, splicing the results, we can get the output feature value s:

$$\text{s} = MultiHead(Q, K, V) = Concat(head_1, \ldots, head_n)W^O \tag{11}$$

3.3 Recommendation Table Generation

This section describes the process of generating the pre-K recommendation tables. We extract the user feature representation through the convolutional neural network, so that we can get the user's potential feature set S_u. Then use the CNN-BLSTM plus the multi-head self-attention mechanism to calculate the set of potential features of the movie S_m. Next, the movie scoring matrix is reduced in dimension using the PCA method [28] to obtain a movie score set S_r.

First, calculate the similarity between the user preference feature and the unrated item $Similarity_1$, and select the top K term as the candidate list C_{k1}, and get the following formula:

$$Similarity_1 = \frac{\left(S_u^i\right)^T S_m^i}{\left\|S_u^i\right\| \cdot \left\|S_m^i\right\|} \tag{12}$$

Then, the similarity between the unrated items and the user's preferences is calculated in turn $Similarity_2$, and the top K items are also selected as the candidate list C_{k2}, and the similarity calculation formula is obtained:

$$Similarity_2 = \left(\frac{\left(S_m^i\right)^T S_m^j}{\left\|S_m^i\right\| \cdot \left\|S_m^j\right\|}\right) + \left(\frac{\left(S_r^i\right)^T S_r^j}{\left\|S_r^i\right\| \cdot \left\|S_r^j\right\|}\right) \tag{13}$$

Among them, S_r^i and S_m^i together constitute the potential representation of project i, S_r^j and S_m^j together constitute the potential representation of project j.

By merging the candidate lists C_{k1} and C_{k2}, the recommendation list C_k and the user relevance $Similarity$ can be calculated by the user preference feature $Similarity_1$ and the user preference average project $\overline{Similarity_2}$. Assuming that the user has q preference items, the similarity can be calculated:

$$\overline{Similarity_2} = \frac{1}{q}\sum\nolimits_{i=1}^{q} Similarity_2 \tag{14}$$

And then, get user relevance:

$$Similarity = \varepsilon \cdot \left(\overline{Similarity_2}\right) + (1 - \varepsilon) \cdot Similarity_1 \tag{15}$$

Where ε is an adjustable parameter. The last calculated $Similarity$ set, select the top K item as the recommended item, and recommend it to the user.

4 Experiments and Analysis

4.1 Dataset and Preprocessing

This article uses MovieLens as the experimental data set and enhances the MovieLens data set. There are several different versions of MovieLens, and we mainly select 1M and 10M data sets. We spliced the ID field in the movies.dat table in the 1M dataset with the 10M dataset in the tags.dat table, which is the newly tagged field in the movies.dat table in the 1M dataset. Add a movie profile field to the movie in the movies.dat table to make it easier to grab features from more dimensions in the movie collection.

Since the length and type of the original data are inconsistent, the data needs to be processed before the fields are entered into the embedded layer.

user.dat table:

- gender field: In the gender field, convert ‘F’ and ‘M’ to 0 and 1.
- age field: Change to a continuous number for each age group (one age range every ten years old).
  movies.dat table:
- type field: First, create a dictionary of text to numbers, because a movie can be of multiple types, so we can convert the categories in type into strings, and store it in a dictionary of numbers, finally, convert the type field of each movie into a list of numbers.
- label field: Same as the type field. The description in the label is converted into a list of numbers.
- introduction field: Unsupervised word vector learning with word2vec [29], the learned word vector is stored in the vocabulary. The word is used as the basic unit of the sentence, and the word is expressed as the corresponding word vector form.

4.2 Experimental Parameters

When training the model, we selected 80% of the MovieLens data set as the training set and the remaining 20% as the test set. In the experiment, we set the learning rate to 0.0001. In the training phase, the filter window size of the convolutional neural network is set to 3, 4, and 5 respectively. To prevent over-fitting, add dropout to the neural network of the text processing module, and dropout is set 0.4. The iteration training number epoch is set to 10 and the mini-batch size is set to 128.

4.3 Experimental Results and Analysis

The experiment uses F1, MAP and NDCG to evaluate user-based collaborative filtering algorithm (UBCF), project-based collaborative filtering algorithm (IBCF), CNN-AT, bidirectional LSTM, deep learning model without multi-head attention, and a deep learning model with the multi-head attention mechanism. A total of six models were used to test the error. The results of the comparative test are shown in Table 1.

Table 1. Recommended performance comparison table

Method	@k								
	5			10			20		
	MRR	MAP	NDCG	MRR	MAP	NDCG	MRR	MAP	NDCG
UBCF	0.4358	0.5172	0.5261	0.4826	0.5738	0.5693	0.5176	0.6183	0.6137
IBCF	0.4571	0.5249	0.5412	0.5107	0.5934	0.5831	0.5318	0.6345	0.6188
CNN-AT	0.5261	0.6017	0.6372	0.5761	0.6402	0.6914	0.6033	0.6619	0.7083
BLSTM	0.5118	0.5977	0.6356	0.5704	0.6387	0.6806	0.5949	0.6581	0.7039
CNN-LSTM	0.5407	0.6284	0.6571	0.5953	0.6596	0.7027	0.6218	0.6817	0.7272
CNN-BLSTM-MHA	**0.5877**	**0.6592**	**0.6844**	**0.6376**	**0.7029**	**0.7108**	**0.6591**	**0.7282**	**0.7443**

As can be seen from Table 1, the overall performance of UBCF and IBCF is poor, mainly due to the lack of understanding of the features. The CNN-AT and the BLSTM model have relatively good effects, and both of them can be well in the entity. The CNN-LSTM model combines the advantages of the two models, so it is slightly better than the general deep learning model. Finally, the CNN-BLSTM model with the multi-head attention mechanism is better than the score prediction results produced by the former. In addition, we also compare the different k values, and found that with the increase of the recommended items, there is a better recommendation effect, and when the recommended items are increased to a certain number, the improvement of the recommendation effect is relatively small.

5 Conclusion and Expectation

In this paper, we propose a user preference learning model that incorporates attention mechanisms to recommend movies to users. We try to use the combination of convolutional neural network, BLSTM and attention mechanism to process user data and movie data, extract features from it, and finally combine movie scores to jointly generate recommendation tables. In the study of attention mechanisms, we also tried to use the multi-head attention mechanism to process and capture features of different subspaces in different locations in parallel, and there is no limit on distance dependence. A comprehensive experimental study demonstrates the effectiveness of our proposed model in film recommendation tasks.

But because our model only considers getting valid information from users and in some dimensions of the movie, it does not consider capturing features from more dimensions, such as movie posters or movie plots. Therefore, in the future we may consider improving our model, considering different features and different models to obtain more effective information in different dimensions, and finally combining to better understand user preferences. Thus providing users with higher quality and more precise recommendations. We will also try to recommend the application of the model in other areas.

Acknowledgement. The authors gratefully acknowledge support from National Key R&D Program of China (No. 2018YFC0831800), National Natural Science Foundation of China (No. 61872134), Natural Science Foundation of Hunan Province (No. 2018JJ2062), Science and technology development center of the Ministry of Education, and the 2011 Collaborative Innovation Center for Development and Utilization of Finance and Economics Big Data Property, Universities of Hunan Province.

References

1. Socher, R., Huval., B., Manning, C.D., Ng, A.Y.: Semantic compositionality through recursive matrix-vector space. In: Conference on Empirical Methods in Natural Language Processing (2012)
2. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751 (2014)

3. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artif. Intell. 1–19 (2009)
4. Zhang, Y., Zincir-Heywood, N., Milios, E.: World wide web site summarization. Web Intell. Agent Syst. Int. J. **2**, 39–53 (2004)
5. Jones, S., Staveley, M.S.: Phrasier: a system for interactive document retrieval using keyphrases. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 160–167. ACM (1999)
6. Socher, R., Perelygin, A., Wu, J.Y., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642 (2013)
7. Tang, D., Qin, B., Wei, F., et al.: A joint segmentation and classification framework for sentence level sentiment classification. IEEE/ACM Trans. Audio Speech Lang. Process. **23**, 1750–1761 (2015)
8. Xie, R., Liu, Z., Sun, M.: Representation learning of knowledge graphs with hierarchical types, pp. 2659–2665 (2016)
9. Jacob, T.I.: A neural model for straight line detection in the human visual system. Dissert. Theses – Gradworks (2014)
10. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. Comput. Sci. (2014)
11. Ramirez-Moreno, D.F., Schwartz, O., Ramirez-Villegas, J.F.: A saliency-based bottom-up visual attention model for dynamic scenes analysis. Biol. Cybern. **107**, 141–160 (2013)
12. Luong, M.-T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation (2015)
13. Xu, J., Chen, K., Qiu, X., et al.: Knowledge graph representation with jointly structural and textual encoding. 1318–1324 (2017)
14. Ding, Y., Li, X.: Time weight collaborative filtering. In: Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany (2005)
15. Su, P., Ye, H.: [IEEE 2009 International Joint Conference on Artificial Intelligence (JCAI) - Hainan Island, China (2009.04.25–2009.04.26)] International Joint Conference on Artificial Intelligence - An Item Based Collaborative Filtering Recommendation Algorithm Usin, pp. 308–311 (2009)
16. Schafer, J.B., K., Konstan, J., Riedi, J.: [ACM Press the 1st ACM conference - Denver, Colorado, United States (1999.11.03–1999.11.05)] Proceedings of the 1st ACM conference on Electronic commerce, - EC 1999 - Recommender systems in e-commerce, pp. 158–166 (1999)
17. Cho, K., Van Merrienboer, B., Gulcehre, C., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation (2014)
18. Cheng, H.T., Koc, L., Harmsen, J., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 7–10. ACM (2016)
19. Zhu, J., Shan, Y., Mao, J., Yu, D., Rahmanian, H., Zhang, Y.: Deep embedding forest: Forest-based serving with deep embedding features. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, USA, pp. 13–17 (2017)
20. Collobert, R., Weston, J., Bottou, L., et al.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
21. Liu, Y., Sun, C., Lin, L., et al.: Learning natural language inference using bidirectional lstm model and inner-attention (2016)
22. Huang, C., Kuo, P.-H.: A deep CNN-LSTM model for particulate matter (PM2.5) forecasting in smart cities. Sensors **18**, 2220 (2018)

23. Wang, X., Yu, L., Ren, K., et al.: Dynamic attention deep model for article recommendation by learning human editors' demonstration (2017)
24. Vaswani, A., et al.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)
25. Zhai, S., Chang, K.-H., Zhang, R., Zhang, Z.M.: Deepintent: learning attentions for online advertising with recurrent neural networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1295–1304 (2016)
26. Zhou, C., et al.: ATRank: an attention based user behavior modeling framework for recommendation. In: AAAI, pp. 4564–4571 (2018)
27. Vaswani, A., et al.: Attention is all you need. CoRR abs/1706.03762 (2017)
28. Dubuisson-Jolly, M.P., Gupta, A.: Color and texture fusion: application to aerial image segmentation and gis updating. Image Vis. Comput. **18**, 823–832 (2000)
29. Mikolov, T., Sutskever, I., Chen, K., et al.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
30. Zhaowei, Q., Cao, B., Wang, X., Li, F., Xu, P., Zhang, L.: Feedback LSTM network based on attention for image description generator. Comput. Mater. Continua **59**(2), 575–589 (2019)
31. Feng, X., Zhang, X., Xin, Z., Yang, A.: Investigation on the Chinese text sentiment analysis based on convolutional neural networks in deep learning. Comput. Mater. Continua **58**(3), 697–709 (2019)
32. Wang, G., Liu, M.: Dynamic trust model based on service recommendation in big data. Comput. Mater. Continua **58**(3), 845–857 (2019)