A long short-term memory deep learning framework for explainable recommendation

Hafed Zarzour[1], Yaser Jararweh[2], Mahmoud M. Hammad[2], Mohammed Al-Smadi[2]

[1] Department of Computer Science
University of Souk Ahras
Souk Ahras, Algeria

[2] College of Computer and Information Technology
Jordan University of Science and Technology
Irbid, Jordan

hafed.zarzour@gmail.com, yijararweh@just.edu.jo, m-hammad@just.edu.jo, maalsmadi9@just.edu.jo

Abstract*— Due to the growing quantity of information available on the Web, recommender systems have become crucial component for the success of online shopping stores. However, most of the existing recommender systems were only designed to improve the recommendation results and ignore the explainable recommendation aspect. Therefore, in this paper we propose a long short-term memory deep learning framework for explainable recommendation, that is able to generate an efficient explanation for any rating made by users for a recommended item. Such a framework would help users to choose a product with confident after reading the automatically generated explanation by our framework. The generated explanation is a concise sentence that shows the reason behind a recommendation, i.e., why a user should select that product. Extensive experiments on a real-world dataset from Amazon are conducted with the goal to evaluate the effectiveness of the proposed method in terms of loss and accuracy metrics. The experimental results demonstrate the effectiveness of our method according to the diversity in generating explainable recommendation.

Keywords*— long short-term memory (LSTM), deep learning, explainable recommendation, recommender system, machine learning.

I. Introduction

Due to the growing quantity of data and information available on the Web, recommender systems have attracted considerable attention in social networks and online shopping websites. Facebook, Twitter, Netflix, and Amazon are the typical examples of such websites. Recommender systems [1] are able to predict which products or items are best suited to particular users, based on their preferences gathered following a specific manner, to perform the recommendation task. These systems are usually divided into two classes: collaborative filtering [2] and content-based filtering [3].

Wang et al. [4] showed that the first class of recommender systems contains one of the most successful approaches in the domain of recommendation. The collaborative filtering class is the widely used in practice because of its feasibility and ease of use. This class includes two main sub-classes: memory-based and model-based techniques. The memory-based techniques use the entire similarities between items or users to produce predictions, whereas the model-based techniques employ the ratings in training a model, which is then used to produce predictions for users' ratings of an unrated item or items.

The model in [5] uses a new algorithm for collaborative filtering recommendations based on dimensionality reduction coupled with a clustering method. A k-means clustering algorithm was used to create clusters of users sharing the same interests and thoughts, while a singular value decomposition was used to establish the dimensionality reduction. In the same way, the authors in [6] proposed two types of collaborative filtering algorithms aiming at improving the prediction accuracy for big data. In the first type, they used a simple k-means clustering algorithm, whereas in the second type they combined the k-means clustering algorithm with the Principal Component Analysis (PCA). Specifically, PCA was adopted as a way to conduct efficient dimensionality reduction with the aim to maintain most of the relevant information before conducting the clustering task. This execution order, dimensionality reduction followed by users clustering, has significantly improved the recommendation results.

Recently, there have been some studies attempting to explore the potentiality provided by the deep learning methods in enhancing the recommendation precision. For example, Zarzour et al. [7] developed a recommender system based on deep neural network called RecDNNing, which aims to predict the rating scores using the algorithm of forward propagation. RecDNNing combines the users and items embeddings with the deep neural network strategy to improve the quality of recommendation in two stages. In the first stage, they created a dens numeric representation for each user and item. After that, they calculated the average and then concatenated the obtained vectors before being fed into the deep neural network. In the last stage, they used the deep neural network model to predict the ratings scores using the forward propagation algorithm.

However, most of the existing recommender systems were designed only to improve the recommendation quality or the prediction accuracy and did not take into account the explainable recommendation aspect. Developing a new generation of recommender systems that can provide an explainable recommendation each time an item is suggested would make users aware of "why" this specific item is recommended.

978-1-7281-6226-3/20/$31.00 ©2020 IEEE

An explanation is a piece of information showed to users, answering the *why-question* about such a recommendation result. For example, instead of knowing which books to read and which products to buy, with explainable recommendations users are able to know *why* books are suggested to read and *why* products are suggested to buy.

Nowadays, the development of explainable recommendations are becoming increasingly attractive direction in the field of information filtering systems. This is to help people around the world not only to take the right decision related to their lives (recommender systems) but also to trust and understand the system they use (explainable recommendation).

Indeed, there are many reasons designers of recommender systems would benefit from such a technology:

- Better understand why items are recommended;

- Make recommendations explainable;

- Increase trust and transparency;

- Improve users' experiences and satisfactions;

- Assist users to make better decision;

- Develop reputation models;

- Sell more products;

- Attract more users.

Therefore, in this paper we propose a long short-term memory (LSTM) deep learning neural network method for explainable recommendations. The proposed method makes not only the recommendations but also generate intuitive textual explanations about the recommended item based on users' ratings. Extensive experiments on a real-world dataset from Amazon are conducted with the goal to evaluate the performance of the proposed approach in terms of training loss and accuracy metrics. The experimental results show the effectiveness of our method according to the diversity in generating recommendation reasons.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 describes the proposed approach. Section 4 illustrates the experimental results. Finally, Section 5 concludes this paper with avenues for future work.

II. Related Work

Unlike traditional recommender systems that recommend items to users, explainable recommendations give the reason why an item is recommended [8][9][10][11].

Explainable recommendations can be defined as explanatory capabilities added to recommender systems for providing trust and transparency to expose the reason behind a recommendation [12][13][14][15]. In another words, explainable recommendations are a class of algorithms that solves the problem related to the question of *why* a particular item is recommended to a given user. We note here that this class of algorithms does not address the old problem of how such systems can provide better recommendation results. A recommender systems integrating explainable recommendation algorithms makes recommendations more reliable and understandable than a simple list of item proposed to a user without any explanation.

Many information sources can be used at the same time to generate explanations with different kinds of presentation such as sentences, videos, and images; implying the existence of several explanations for the same recommendation results.

Following are some examples of the most recent related studies. Lin et al. [16] developed a neural network framework that is capable to generate abstractive comments when providing outfit recommendations. They employed a convolutional neural network (CNN) with a mutual attention function to outfit matching and a recurrent neural network (RNN) with a cross-modality attention function to obtain a concise sentence. Ai et al. [17] provided a knowledge-base embeddings framework for explainable recommendation in which they used a soft matching algorithm to provide explanation about the recommended items. The soft matching algorithm was divided for exploring the explanation paths that exist between the recommended items and users.

Hou et al. [18] proposed an approach to generate explainable recommendations using aspect-based matrix factorization, which could enhance the prediction of rating with fusion of aspect information. Finally, Ribeiro et al. [19] proposed an approach called LIME, an algorithm that is capable of explaining the predictions of any classifier in a faithful way. They also introduced SP-LIME to filter the representative predictions from those redundant and non-representative ones.

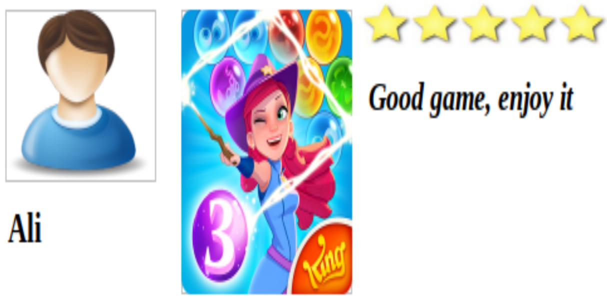
III. Our Approach

In this study, we develop a long short-term memory (LSTM) deep learning neural network framework for explainable recommendations that aims not only to make accurate recommendations for users but also to provide them with the reasons for these recommendations. Such a framework would provide users using recommender systems a transparent explanation mechanism for each recommended item as they receive recommendations from their peers. The long short-term memory deep learning framework for explainable recommendation consists of three main parts, which are (1) the input gate for ratings and reviews data, (2) the long short-term memory model, and (3) the generated explainable recommendations. The first part describes the input of our model, while the last part describes its output that will be displayed to users.

A. Input gate for ratings and reviews

The input data for our model contains two vectors: the ratings and the corresponding reviews. The rating is usually extracted from the user-item ratings matrix and used to model the relationship between items and users. It is a numerical value given by a user to an item in order to express explicitly how much he likes that item. More formally, the rating score for a given item *I* made by user *U* is described as follows:

$R_{i,j}(U_i, I_j)$, where $R_{i,j}$ is the rating, U_i is the ith user, and I_j is the jth item. The user and item can be represented by their



ID_u and ID_i , respectively.

Fig. 1. Example of a user and an item with their star rating and a review.

On the other hand, a review is a textual description of an item *I* provided by a user *U*. For example, if the item is a game called "Bubble Witch 3 Saga", an interpretation of this game can be a short review like "Good game, enjoy it" with five stars as a rating given by the user Ali as in the example given in Figure 1.

We use both ratings and reviews vectors to generate automatic reasons explaining why a particular item *I* is recommend to an active user.

B. Long short-term memory model

A LSTM [20] is a kind of recurrent neural network (RNN) architectures that overcomes some of the limitations faced in deep recurrent neural networks. More specifically, the vanishing and exploding gradients [21] are considered the most severe issues occurred during the training stage deep neural networks, especially when taking long range dependencies in the data.

The model of LSTM uses the memory cell concept with self-connections to store information as temporal state as well as input and output gates to manage the information flow. These gates and memory cell construct the memory cell block of the model.

A sequence X= $(x_1,x_2,...,x_n)$ is taken as input and returned as output in another sequence Y=$(y_1, y_2...,y_m)$ by computing the activation of the network according to the following implementation[22][23]:

$$i_t=\sigma(W_{xi}X_t+W_{hi}h_{t-1}+W_{ci}c_{t-1}+b_i) \qquad (1)$$

$$f_t=\sigma(W_{xf}X_t+W_{hf}h_{t-1}+W_{cf}c_{t-1}+b_f) \qquad (2)$$

$$c_t=f_t c_{t-1}+i_t\, tanh(W_{xc}X_t+W_{hc}h_{t-1}+b_c) \qquad (3)$$

$$o_t=\sigma(W_{xo}X_t+W_{ho}h_{t-1}+W_{co}ct+b_o) \qquad (4)$$

$$h_t=o_t\, tanh(c_t) \qquad (5)$$

where i_t is the input gate, f_t is the forget gate, c_t is the cell state, o_t is the output gate, h_t is the hidden layer, X_t is the current sample, and W_{xi} denotes the weight matrix.

C. Explainable recommendations

The explainable recommendation is the output of our model, in which the reasons interpreting the recommendation is personalized from a couple of other reviews about the target item. More typically, the generated explainable recommendation is a sequence of words (Z_1, Z_2, Z_3...Z_n) that is readable, understandable , and coherent with the rating.

IV. Experiments

In this section, we explain our experiments on a real-world dataset to evaluate the performance of the proposed approach. The dataset called Amazon Product Data (http://jmcauley.ucsd.edu/data/amazon/) from Amazon, which includes millions of product reviews (e.g., ratings and text) and metadata (e.g., description, price, and category information).

Since in this study we are not interested in solving the traditional recommender systems' problems, we focus on the 5-core subset in which only the items and users having at least five reviews are considered. To conduct our experiments, we used the reviews from the "Toys and Games" category from the dataset. We only considered the reviews that are less than 50 characters long. "Fun and educational", "Great 1st birthday gift!" , "Good product for babies" , and "kids love it" are some examples of reviews considered in our experiments after excluding the long reviews.

To assess the effectiveness of the proposed approach, the training loss and the accuracy are computed after each epoch with three different diversity values when generating the explainable recommendations. In this work, the diversity corresponds to the diversity in the generated recommendations. Therefore, the performance of the system is measured three times with diversity = 0.5, 1.0, and 1.5, respectively. The used loss function is represented as follows:

$$\sum_0^n p_i \log(\frac{1}{q_i}) \qquad (6)$$

where p_i denotes the ground truth probability and q_i denotes the predicted probability after using our model.

Fig. 2. Training loss for our model with three diversity values.

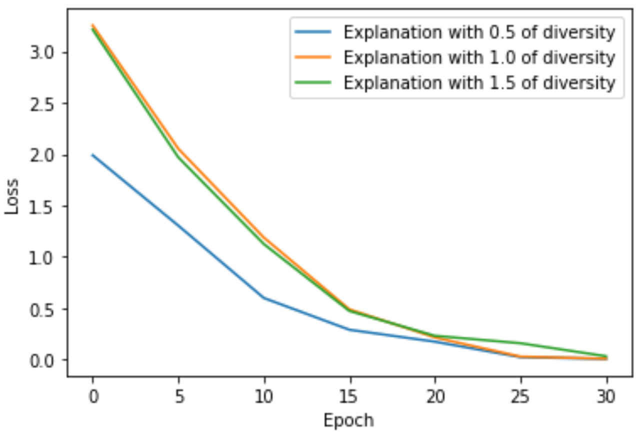


Figure 2 depicts the gradual degradation in the values of loss for our model using different diversity values achieved at 30th epoch of training. It can be observed that the loss values are degraded for all diversity values.

In particular, the loss values are smallest in the case of the model having the diversity =0.5, the blue lower line, and largest in the case of the diversity =1.0 and 1.5, the two upper lines. Thus, the best performances of the proposed model with different diversities are achieved when diversity =0.5, while great diversity values do not help to generate new explainable recommendations.

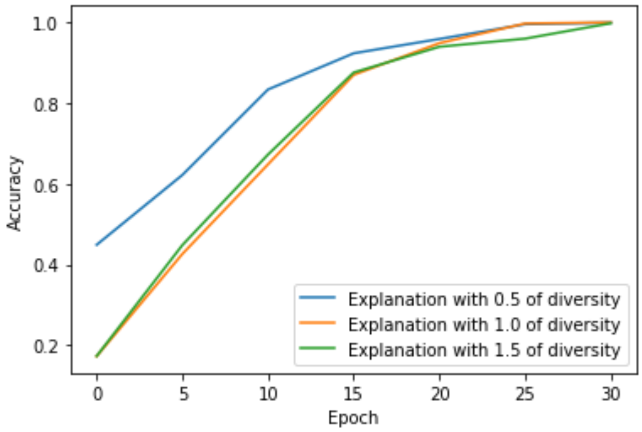


Fig. 3. Accuracy for our model with three diversity values.

Figure 3 depicts the learning accuracy of our model with three diversity values achieved at 30^{th} epoch. It can be noted that for all diversity values the accuracy increases as the number of epochs increases, especially after the 15^{th} epoch.

The presented experimental results in this section demonstrate the effectiveness of the proposed approach when the diversity = 0.5 in the automatic generation of reasons linked to recommendations.

From Figures 2 and Figure 3, it can be concluded that the performance of our model is based on the diversity in generating reasons for explainable recommendations. In addition, it can be inferred that our system with 0.5 diversity can perform better as it generates more explainable texts by showing a gradual degradation in the training loss with increasing accuracy.

V. Conclusion and Future Work

In this paper, we have proposed a long short-term memory (LSTM) deep learning neural network method for explainable recommendations. To that end, we combined a couple of variables that consists of ratings and reviews as input data with an LSTM model as a deep learning model to generate accurate explanations in order to make recommendations better understandable by clarifying why a particular item is recommended.

We conducted comprehensive experiments on a real-world dataset from Amazon and the corresponding results indicated that our approach can generate useful reasons for any performed recommendation. The performance of our approach is linked to one key decision, which is the diversity value. Thus, the diversity in the reasons generations for explainable recommendations is the most important factor that has a direct impact on the performance of our model. This research study confirms that the LSTM model is a well-suited model to be applied for providing explainable recommendations in recommender systems.

In future, we will explore more advanced deep learning methods to further enhance the quality of explainable recommendations. Moreover, we will conduct more experiments on additional datasets to compare the performance of our model with other state-of-the-art models.

References

[1] F. Maazouzi, H. Zarzour, and Y. Jararweh, “An Effective Recommender System Based on Clustering Technique for TED Talks,” International Journal of Information Technology and Web Engineering, vol. 15, no. 1, pp. 35–51, Jan. 2020.

[2] P. Thakkar, K. Varma, V. Ukani, S. Mankad, and S. Tanwar, “Combining User-Based and Item-Based Collaborative Filtering Using Machine Learning,” Smart Innovation, Systems and Technologies, pp. 173–180, Dec. 2018.

[3] Y. Ham and M. Kamari, “Automated content-based filtering for enhanced vision-based documentation in construction toward exploiting big visual data from drones,” Automation in Construction, vol. 105, p. 102831, Sep. 2019.s

[4] K. Wang, L. Xu, L. Huang, C.-D. Wang, and J.-H. Lai, “SDDRS: Stacked Discriminative Denoising Auto-Encoder based Recommender System,” Cognitive Systems Research, vol. 55, pp. 164–174, Jun. 2019.

[5] H. Zarzour, Z. Al-Sharif, M. Al-Ayyoub, and Y. Jararweh, “A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques,” 2018 9th International Conference on Information and Communication Systems (ICICS), Apr. 2018.

[6] H. Zarzour, F. Maazouzi, M. Soltani, and C. Chemam, “An Improved Collaborative Filtering Recommendation Algorithm for Big Data,” IFIP Advances in Information and Communication Technology, pp. 660–668, 2018.

[7] H. Zarzour, Z. A. Al-Sharif, and Y. Jararweh, “RecDNNing: a recommender system using deep neural network with user and item embeddings,” 2019 10th International Conference on Information and Communication Systems (ICICS), Jun. 2019.

[8] B. Abdollahi and O. Nasraoui, “Explainable Matrix Factorization for Collaborative Filtering,” Proceedings of the 25th International Conference Companion on World Wide Web - WWW ’16 Companion, 2016.

[9] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, “Explicit factor models for explainable recommendation based on phrase-level sentiment analysis,” Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval - SIGIR ’14, 2014.

[10] N. Wang, H. Wang, Y. Jia, and Y. Yin, “Explainable Recommendation via Multi-Task Learning in Opinionated Text Data,” The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval - SIGIR ’18, 2018.

[11] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, “Explainable Reasoning over Knowledge Graphs for Recommendation,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5329–5336, Jul. 2019.

[12] H. Hou and C. Shi, “Explainable Sequential Recommendation using Knowledge Graphs,” Proceedings of the 5th International Conference on Frontiers of Educational Technologies - ICFET 2019, 2019.

[13] J. Gao, X. Wang, Y. Wang, and X. Xie, “Explainable Recommendation through Attentive Multi-View Learning,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3622–3629, Jul. 2019.

[14] Z. Chen, X. Wang, X. Xie, T. Wu, G. Bu, Y. Wang, and E. Chen, “Co-Attentive Multi-Task Learning for Explainable Recommendation,” Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Aug. 2019.

[15] N. Tintarev and J. Masthoff, “Designing and Evaluating Explanations for Recommender Systems,” Recommender Systems Handbook, pp. 479–510, Oct. 2010.

[16] Y. Lin, P. Ren, Z. Chen, Z. Ren, J. Ma, and M. De Rijke, “Explainable Outfit Recommendation with Joint Outfit Matching and Comment Generation,” IEEE Transactions on Knowledge and Data Engineering, pp. 1–1, 2019.

[17] Q. Ai, V. Azizi, X. Chen, and Y. Zhang, “Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation,” Algorithms, vol. 11, no. 9, p. 137, Sep. 2018.

[18] Y. Hou, N. Yang, Y. Wu, and P. S. Yu, “Explainable recommendation with fusion of aspect information,” World Wide Web, vol. 22, no. 1, pp. 221–240, Apr. 2018.

[19] M. Ribeiro, S. Singh, and C. Guestrin, “Why Should I Trust You?’: Explaining the Predictions of Any Classifier,” Proceedings of the

2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, 2016.

[20] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[21] B. Hanin, “Which neural net architectures give rise to exploding and vanishing gradients?,” In Advances in Neural Information Processing Systems, pp. 582-591, 2018.

[22] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” In Fifteenth annual conference of the international speech communication association, 2014.

[23] M. Abdel-Nasser and K. Mahmoud, “Accurate photovoltaic power forecasting models using deep LSTM-RNN,” Neural Computing and Applications, vol. 31, no. 7, pp. 2727–2740, Oct. 2017.