IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication: 1 January 2021

Digital Object Identifier 10.1109/ACCESS.2021.DOI

Improving explainable recommendations by deep review-based explanations

SIXUN OUYANG[1], AONGHUS LAWLOR[2]
[1]Insight Centre for Data Analytics, University College Dublin, Ireland (e-mail: sixun.ouyang@insight-centre.org)
[2]Insight Centre for Data Analytics, University College Dublin, Ireland (e-mail: aonghus.lawlor@insight-centre.org)

Corresponding author: Sixun Ouyang (e-mail: sixun.ouyang@insight-centre.org).

This work is supported by Science Foundation Ireland through the Insight Centre for Data Analytics under grant number SFI/12/RC/2289_P2

ABSTRACT Many e-commerce sites encourage their users to write product reviews, in the knowledge that they exert a considerable influence on users' decision-making processes. These snippets of real-world experience provide an essential source of data for interpretable recommendations. However, current methods relying on user-generated content to make recommendations can run into problems because of well-known issues with reviews, such as noise, sparsity and irrelevant content. On the other hand, recent advances in text generation methods demonstrate significant text quality improvements and show promise in their ability to address these problems. In this paper, we develop two character-level deep neural network-based personalised review generation models, and improve recommendation accuracy by generating high-quality text which meets the input criteria of text-aware recommender systems.To make fair comparisons, we train review-aware recommender systems by human written reviews and attain advanced recommendations by feeding generated reviews at the inference step.Our experiments are conducted on four large review datasets from multiple domains. We leverage our methods' performance by comparing with non-review based recommender systems and advanced review-aware recommender systems. The results demonstrate that we beat baselines on a range of metrics and obtain state-of-the-art performance on both rating prediction and top-N ranking. Our sparsity experiments validate that our generation models can produce high-quality text to tackle the sparsity problem. We also demonstrate the generation of useful reviews so that we can achieve up to 13.53% RMSE improvements. For explanation evaluation, quantitative analyses reveal good understandable scores for our generated review-based explanations, and qualitative case studies substantiate we can capture critical aspects in generating explanations.

INDEX TERMS Computing methodologies, Deep neural networks, Information systems, Natural language generation, Recommender systems

I. INTRODUCTION

RECENT recommender systems research has shown how to make use of online user-generated content to continuously improve performance on both recommendation [1]–[5] and explanation [6]–[8]. However, this approach has two main drawbacks. First, the problem of sparse data continues to present an issue for most recommender systems [9]. When there is sparse rating information for users and items, which is often the case for review-aware recommender systems [8], the latent factor-based methods often struggle to learn significant information and can produce inaccurate results. Second, there is the problem that a large fraction of the content of human-written reviews is not useful information for the recommender system. According to Chen *et al.* [5], user-written reviews are of low quality that and unhelpful for obtaining users' trust and producing accurate recommendations. They also argued that these unhelpful reviews only add noise which impairs the ability of the recommender system. Besides, Ghose *et al.* [10] researched the subjectivity of online reviews and concluded that users prefer helpful reviews which introduce object information about items rather to express subjective prejudices. Meanwhile, Jindal *et al.* [11] analysed 5.8 million reviews from Amazon [1] and argued that opinion spam in online reviews is extensive. We demonstrate two examples (rating 4 and rating 1) of unhelpful reviews from Amazon in Figure 1. The first example does not present

[1]https://www.amazon.com/

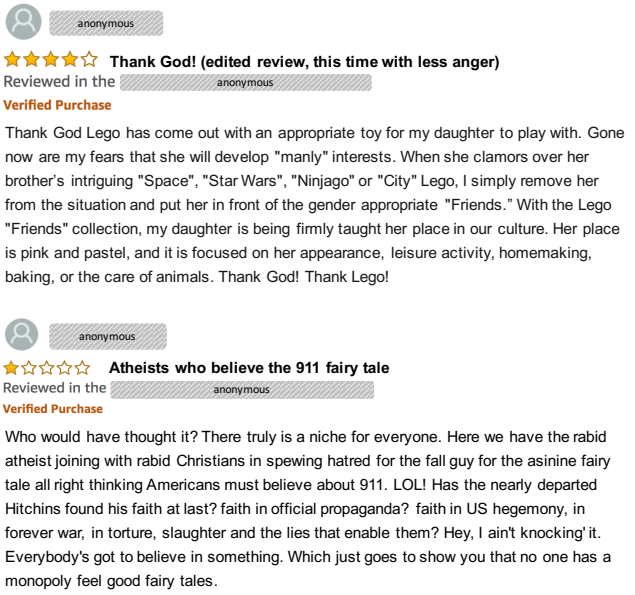


FIGURE 1: Unhelpful online reviews from Amazon.

useful information about the item but just expresses gender discrimination, so that it is marked as unhelpful. The second example shows spam opinion and spreads extreme political ideologies, which is not profitable for recommendations and explanations.

On the other hand, there has been intense interest in high-quality text generation through deep learning methodologies. Recent works demonstrate that deep neural network-based text generation models can generate high-quality synthetic text with correct grammar and syntax [7], [12]–[14]. Meanwhile, they argue that machine generated reviews are compelling and useful. Therefore, many works now integrate text generation methods with recommender systems for constructing explainable recommendation [15]–[21]. These results have validated the idea that artificially generated review text is good for explaining recommendations. However, the use of machine-generated review explanations to improve recommendation performance has not been explored in detail. According to Loyola *et al.* [22], explanations can be of a similar format to the input data, so that they can be used to refine predictions. To this end, it is valuable to study how well review-based explanations can work in improving recommendations, on both rating prediction and top-N ranking tasks.

In this paper we propose deep learning review generation models for high-quality natural language explanations and we use these explanations to reach state of the art performance on rating prediction and top-N item ranking recommendation tasks. Specifically, we develop two character-level personalised review generation models which we apply to generate useful reviews for recommendation and explanation. We apply Long Short Term Memory (LSTM) [7] neural network as the basis for our generation models and conduct cross-domain experiments on two Convolutional Neural Network (CNN) based recommender systems [3], [5]. Our experiments show that we beat baselines on both rating prediction and item ranking problem. We demonstrate that machine-generated reviews are more robust than human-written reviews in dealing with sparsity issues. We validate the assumption that our generation models focus on generating useful instead of non-helpful reviews, which boosts the performance of the recommender system. Besides, our explanation experiments demonstrate that we can produce both persuasive and readable explanations for recommendations. We summarise our contributions as follows:

- We develop two character-level personalised review generation models consistently outperforming baselines on both rating prediction and item ranking tasks.
- We show that machine-generated reviews can mitigate the impact of sparsity for text-aware recommender systems.
- Regarding recommendation, we demonstrate the generated machine-generated reviews are more reliable and helpful than human-written reviews.
- We provide convincing and readable text explanation for the predicted ratings.

In following sections, we start to introduce the background of deep review-based explanation generation. Then we outline the proposed approach of using the generation model to improve recommendation accuracy. After that, we introduce the technical details of our explanation generation models. We then analyse the recommendation and explanation experimental results. Finally, we summarise our findings and contributions in conclusion.

II. RELATED WORK

There is a long history of using Natural Language Processing (NLP) methods to exploit cross-domain information for recommendations. Mooney *et al.* [23] presented the first research to extract text information applied to the problem of book recommendation. Van *et al.* [24] employed *tf-idf* schemes to explore text. In order to extract extensive information, many recent works used machine learning and deep learning methods. Based on the idea of word embedding model, Grbovic *et al.* [1] proposed prod2vec model delivering effective product recommendations. Chen *et al.* [4] showed reliable performance on graph-based recommendations by exploring social tag info through Latent Dirichlet Allocation (LDA). Almahiri *et al.* [25] enhanced recommendation performance through Recurrent Neural Networks (RNNs) learning context of reviews. In addition to RNNs, convolutional neural networks (CNNs) are also popular approaches for modelling review text and achieving substantial improvements in recommendation performance [3], [5], [26], [27].

In addition, recent research has focused on the approach of explaining recommendations by natural language-based explanations [28], [31]–[33]. Chang *et al.* [29] presented an explanatory model, learning tags from users' reviews, and

Explanation Methods	Advantages	Disadvantages
Aspect Forecasting [5], [28]–[33]	1. Can predict explainable aspects efficiently. 2. Sensitive to changes of preference. 3. Can map from user needs to recommendations. 4. Easy to produce explanations.	1. Present in predefined templates. 2. Lack of attractiveness and persuasiveness. 3. Must define the aspects dictionary in advance. 4. Can not improve recommendation.
Automatic Natural Language Generation [7], [12]–[14], [18], [34]–[36]	1. Present in readable natural language expression. 2. Good attractiveness and persuasiveness. 3. No need to extract phrases in advance. 4. Best-in-class performance on text generation. 5. Can improve recommendation.	1. Requires a large amount of data. 2. Requires extensive fine-tuning skills. 3. Extremely computationally expensive to train. 4. Opaque prediction process.

TABLE 1: Advantages and disadvantages of aspect explanation and natural language explanation.

filling the predicted tags into a template for structured interpretations. Musto *et al.* [30] introduced a similar approach which uses a textual template with predicted properties to explain recommendations. Wang *et al.* [33] introduced a multi-task learning model for explainable recommendations, where their model can directly predict opinionated phrases. Explanation through these aspect-level approaches is often not entirely adequate for convincing customers. The research of Costa *et al.* [7] indicated that user-written reviews significantly affects other users' purchasing behaviour. Moreover, Seo *et al.* [26] provided a human-understandable interpretation approach which highlights components in reviews.

Based on the above literature, human-written reviews play a critical role in both recommendations and explanations. However, using them directly can run into problems. According to the work of Wan *et al.* [8], human-written reviews among all purchasing records in many real-world datasets are very sparse, for example, only *2.2%* records contain reviews in the Steam dataset [37]. Review-aware recommender systems often struggle to capture significant knowledge and their performance suffers when there are few records for users and items [9]. Also, the content of human-written reviews may not always useful for explainable recommender systems. Chen *et al.* [5] argued that these unhelpful human written-reviews add noise and limit the effectiveness of the recommendations. For these reasons we argue that carefully crafted machine-generated reviews can be a better choice than human-written reviews for building better recommendation systems and providing compelling explanations to users.

Early attempts at automatic text generation used aggregation rules on words to construct sentences [38]. Recently, many text generation works using deep learning methods show significant improvements in generating text that can be readily understood by humans [7], [13], [34]–[36]. The methods are mainly based on RNNs, a type of feed-forward recurrent neural network, which specialise in dealing with sequential tasks. Intrinsically, text generation is a sequence prediction problem, in which the RNNs learns the patterns within the text and predicts the current word (or character) from previous words (or characters). Sutskever *et al.* [34] was the first to introduce a model that used RNNs for synthetic text generation, while other works have greatly expanded it. Tang *et al.* [35] extended a multi-layer perceptron to RNNs which captures personal information and the reduces memory cost. Dong *et al.* [13] further developed this model by using attention mechanisms and demonstrated superior text generation performance. Zhao *et al.* [14] proposed a generation model producing readable text for explaining song recommendation, which is the most similar work to this paper. Similarly, Avinesh *et al.* [18] also proposed text summarisation model based on encoder-decoder sequence networks to summary reviews as recommendation explanation. Nonetheless, these works generate text at word-level and a big issue they face is a high memory cost due to the large vocabulary size. On the contrary, generating text at character-level does not suffer from this problem [12]. Although it is more difficult than word-level generation, this paper focuses on machine-generated reviews at character-level. Table 1 compares the above aspect explanation works and natural language explanation works.

The advantages of machine-generated reviews are considerable. High quality, focused, and readable text can be provided for any item. The quality of the text is at a high level of readability. There is extensive interest in using human-generated text to improve recommendation models, but no related research has been done on how synthetic text generation can improve recommendation performance. Therefore, we aim to develop personalised text explanation generation models through deep learning methods, generating review-based explanations and using such explanations as inputs of review-ware recommender systems to achieve state-of-the-art recommendation performance.

III. PROBLEM STATEMENT

Our first research question is how to generate reviews which fulfill the demands of a review aware recommender systems? We do this by building text generation models that can provide personalised readable reviews. We design two character-level personal review generation models by using deep neural techniques, inspired by recent text generation works [7], [12], [13], [35]. There are two main advances in our approaches over other generation methods. Firstly, we generate reviews on a character level, which is more stringent than word level models; secondly, our target is not just to

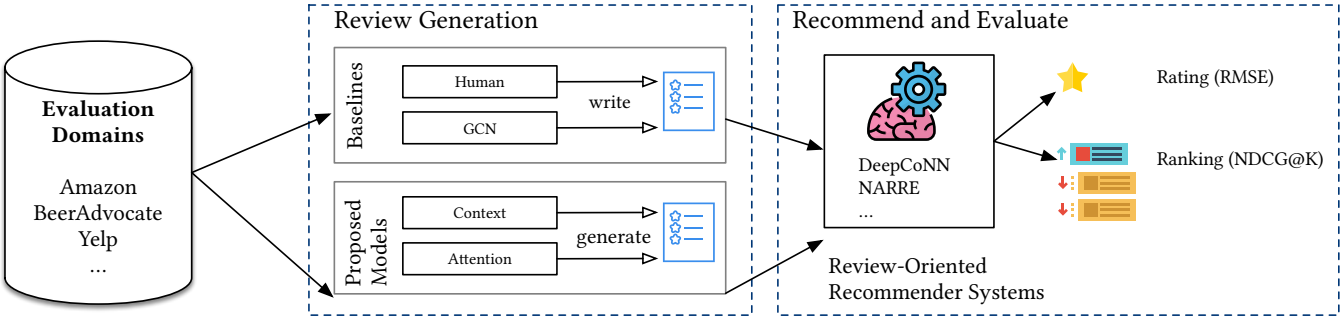


FIGURE 2: Overview of the experimental setup to validate recommendation performance of machine-generated reviews.

generate reviews, but rather to utilise generated reviews to make improvements in the precision of the recommendations and to provide explanations for the recommendation.

The second research problem is how to improve the recommendation performance through machine-generated reviews? To address this, we describe the processes of improving recommendations by our methods and related evaluation metrics, as shown in Figure 2. Concretely, we train review-aware recommender systems by human-written reviews and make advanced recommendations by inputting machine-generated reviews. The idea is that if machine-generated reviews can achieve better performance on recommender systems trained by human-written reviews, we can conclude that deep generation models can capture more meaningful information for producing novel recommendations.

Moreover, to evaluate whether machine-generated reviews can achieve more accurate recommendations or not and measure how they make the improvements, we take the recommendations by human-generated reviews as the ground truth baseline. We run two recommendation tasks, rating prediction and item ranking, which are frequently used in real-world systems. We adopt Root Mean Square Error (RMSE) to estimate rating prediction performance and use NDCG@K to leverage ranking performance. We outline the technical details of our generation models in the next section and detail our experimental results in Section V.

IV. REVIEW GENERATION MODELS

In this section, we provide a detailed overview of our models along two distinct branches: context and attention. Note, the *attention model* is an advanced version of the *context model*, which adds an extra layer with an attention mechanism. Figure 3 demonstrates the neural network architecture of the attention model. There are four modules in the attention model: encoder, Recurrent Neural Networks (RNNs) decoder, attention mechanism, and review generation. The function of the encoder module is to encode attributes and text into high dimensional embeddings; the function of the RNN decoder is to learn the sequence and personal attributes from the embedded inputs; The attention mechanism is used to reinforce the alignment between input attributes and the text; The review generation module generates personalised reviews. The attention model inserts an attention mechanism

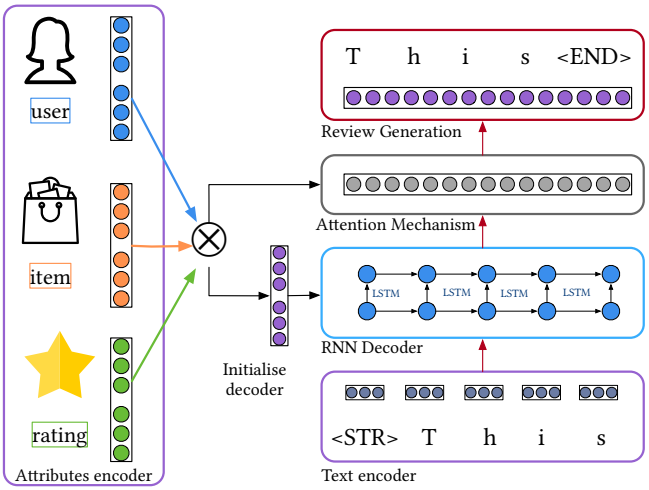


FIGURE 3: Text generation models. The attention model inserts an attention mechanism between the RNNs decoder and text generation module, while the context model directly stacks the text generation module on the RNNs decoder.

between the RNNs decoder and review generation module, while the context model directly stacks the review generation module on RNNs decoder.

In terms of training, the goal of the generation model is to maximise the conditional probability $p(e|a)$ of characters in generated text e where a is the user attributes. To achieve that, our models minimise the cross-entropy loss [7], which is formulated in Equation 1. Here, l represents the sequence length, o and p denote the target characters and predicted characters respectively.

$$p(e|a) = \prod_{t=1}^{l} p(y_t|y_{<t}, a)$$
$$\mathcal{J} = -\sum_{t=1}^{l} [o_t \log y_t + (1 - o_t) \log (1 - y_t)] \quad (1)$$

A. ENCODER

The encoder module aims to encode inputs. It is categorised into two sections by the input type: text and personalised

attributes. The input text in our model is represented by a sequence of characters, while the personalised attributes consist of *user ID*, *item ID*, and *rating*.

To encode the training review text, we first create a dictionary for all unique characters in the training corpus to record their positions. We use this in this encoding process and the later generation module. Then, for each character, we use their index in the dictionary to encode them into a one-hot vector whose length equals the size of the created dictionary. After that, these vectors are then fed into the RNNs decoder directly.

For each attribute, the initial step of encoding is the same with text encoding. We apply a one-hot vector to represent the current attribute. Then, we design a multi-layer perceptron with one hidden layer to linearly transform the one-hot vector into an embedding with a fixed dimension. Specifically, when receiving a one-hot representation $e(a_i)$, where $i \in (1, \dots, |a|)$, the formulation of attribute encoding is shown in Equation 2, where $W_i^a \in \mathbb{R}^{m \times |a|}$ is a weighting matrix, m denotes the dimension of the encoded embeddings, $|a|$ stands for the number of attributes.

$$E(a_i) = W_i^a e(a_i) + b_i^a \tag{2}$$

To align with text, we concatenate the attributes' embeddings and feed them into a fully connected multi-layer perceptron that is activated by a $\tanh$ function. This fully connected layer outputs a hidden vector with the same dimension as the weights of RNNs decoder. We use this hidden vector to initialise RNNs decoder so that the model can be personalised. We define this procedure in Equation 3, where H is a parameter matrix, and b_a denotes the bias.

$$A = \tanh(H[E(a_1), \dots, E(a_{|a|})] + b_a), \tag{3}$$

B. DECODER

RNNs are feed-forward networks with dynamic temporal behaviour aiming to process and learn sequential data and are commonly applied in most text generation systems [7], [12], [13], [35]. Regarding the text generation task, RNNs summarise the context information into hidden variables and then provide conditional probability distributions for each time step. In the vanilla RNN, given an input vector X_t during a time step t and the cell state of previous time step $t-1$, it performs a $\tanh$ activation to get a hidden state h_t of time t. The prediction $p(y_t|y_{<t}, a)$ of time t is calculated by passing the hidden state to an output layer activated by a non-linear $softmax$ function, as shown in Eq. 4.

$$\begin{aligned} h_t &= \tanh(W_x X_t + W_h h_{t-1}) \\ p(y_t|y_{<t}, a) &= \text{softmax}(W h_t + b) \end{aligned} \tag{4}$$

This mechanism enables conventional RNNs to learn the sequential contexts in the input data. However, suffer a few well-known issues including the gradient vanishing problem and to solve this issue, Hochreiter *et al.* [39] introduce the long short-term memory (LSTM) cells, consisting of a set of gates: forget f, input i, and output o. The forget gate decides to discard useless information of input data. The input gate aims to remember decisive information of input data. The output gate determines which information can be passed to the next neuron and the next layer. Unlike vanilla RNNs, forward calculations of an LSTM unit involves inputs x_t, cell state C_{t-1} from the previous unit, and previous unit output H_t. Formulated calculation steps are defined in Equation 5, where W and b stand for weights and bias respectively, $\hat{C}$ is candidate cell state, and $\odot$ denotes the element-wise operator.

$$\begin{aligned} \hat{C}_t &= \tanh(W_x^c x_t + W_h^c H_{t-1} + b_c) \\ f_t &= \sigma(W_x^f x_t + W_h^f H_{t-1} + b_f) \\ i_t &= \sigma(W_x^i x_t + W_h^i H_{t-1} + b_i) \\ C_t &= f_t \odot C_{t-1} + i_t \odot C_t' \\ o_t &= \sigma(W_x^o x_t + W_h^o H_{t-1} + b_o) \\ H_t &= o_t \odot \tanh(C_t) \end{aligned} \tag{5}$$

C. ATTENTION MECHANISM

Attention mechanism shows promising performance on time series related tasks [40], [41]. We use the attention mechanism to prevent the model from concentrating on unrelated information. When receiving outputs H_t of the RNN decoder and encoded attributes $E(a)$ in time t, we first compute context vector G_t through:

$$\begin{aligned} s_t^i &= \frac{\exp(\tanh(W_s[H_t, E(a_i)]))}{\sum_{i=1}^{|a|} \exp(\tanh(W_s[H_t, E(a_i)]))} \\ G_t &= \sum_{i=1}^{|a|} s_t^i E(a_i) \end{aligned} \tag{6}$$

Here, $[,]$ denotes a concatenation manipulation, $|a|$ means the total number of attributes, s_t^i is the attention weights of attributes i in time t. After that, we calculate a new decoder representation $\hat{H}_t$ which has same shape as the RNN decoder outputs by Equation 7, where W is the weight of this layer. In the attention model, $\hat{H}_t$ replaces H_t in generation step.

$$\hat{H}_t = \tanh(W_g G_t + W_h H_t) \tag{7}$$

D. REVIEW GENERATION

Text generation is described as a sequence label classification problem. When accomplishing the encoding and decoding process, we deliver the output H_t from the decoder, or from the attention mechanism, into the output layer, a fully connected neural network with $Softmax$ activation, to compute the conditional probabilities $p(y_t|y_{<t}, a)$. Then, the generation module maximises the conditional probabilities $p(y_t|y_{<t}, a)$ by a greedy search function to predict the character index Y_t. Finally, we generate a character by looking up Y_t in the dictionary created previously. This procedure is applied recursively, and a group of characters will be generated until we find the pre-defined end symbol. The

Dataset name	Users	Items	Ratings	Sparsity (%)
Video	5,130	1,683	37,126	99.57
Beer	22,801	20,200	528,870	99.88
Toys	19,412	11,924	167,597	99.92
Yelp	1,326,101	174,567	5,261,668	99.99

TABLE 2: Statistics of the evaluation datasets

calculation steps of this procedure are presented in Equation 8:

$$p(y_t|y_{<t}, a) = \text{softmax}(W H_t + b)$$
$$Y_t = \text{argmax}\, p(y_t|y_{<t}, a) \quad (8)$$

V. EXPERIMENTS

In this section, we perform extensive experiments through the processes detailed in Figure 2, and discussed in Section III, to answer the following questions:

- Q1 How does the performance of generation models compare to state-of-the-art competitors on both rating prediction and items ranking tasks?
- Q2 Can we produce high-quality review-based explanations to tackle the sparsity problem?
- Q3 Does machine generated review-based explanations are helpful for both recommendations and explanations?
- Q4 How does machine generated review-based explanations perform in quantitative and qualitative study of explanation evaluation?

A. EXPERIMENTAL SETTINGS

1) Datasets

In our experiments, we employ four datasets from different realms: two Amazon 5-score[2] [42]: *Video* and *Toys*; *Beer* [43]; and *Yelp* challenge 2017[3]. Statistical details of evaluation datasets are introduced in Table. 2. Similar to most generation tasks [7], [13], we preprocess datasets as follows: (i) we filter the reviews whose lengths are greater than 512 characters, as suggested in Dong *et al.* [13], long reviews often focus on describing irrelevant information, while short reviews tend to concentrate on more relevant information to the user experience of the item; (ii) since both generation models and review aware recommender systems require adequate numbers of reviews to train, we split each dataset into *generation train*, *recommendation train*, *validate*, and *test* set in the proportion of 40%, 40%, 10% and 10% respectively. Here, the *validate* set is used to select the best hyper-parameters for both the generation model and recommendation model, and the *test* set is used to evaluate recommendation and explanation performance for both machine generated reviews and human written reviews. The advantage for this strategy is that we ensure fair comparison. Generation models are trained on *generation train* set, while recommender systems are trained on *recommendation train* set. In this way, the generated reviews will not contain the information used in training the recommendation system. Thus, we can compare with human written reviews equitably.

[2]http://jmcauley.ucsd.edu/data/amazon
[3]https://www.yelp.com/dataset

Non-review aware recommender systems	Video	Toys	Beer	Yelp
NMF	1.244	1.199	1.638	1.052
SVD	0.986	0.932	1.175	0.964
SVD++	0.980	0.926	1.181	0.965

TABLE 3: RMSE Performance for non-review aware recommender systems (RMSE)

2) Reproducibility

We implement our generation models and related recommender systems based on Tensorflow[4]. Similar to [13], we stack two RNNs layers in the generation models to generate reviews, where each layer contains 512 LSTM neurons. To capture personalisation, we use dimensions of 64 for each attribute. The weights are initialised from a uniform distribution in the range of $[-0.08, 0.08]$ as suggested by [44]. We apply Adam optimisation [45] tuning models with an initial learning rate of 0.002 and unrolling for 50 epochs. According to [46], to avoid over-fitting, we decrease the learning rate after every epoch by multiplying with a factor of 0.97 and stack a dropout layer on each hidden layers with a dropout probability of 0.2. Then, we clip gradients in a range of $[-5, 5]$ to avert the gradient exploding problem [47]. Since short reviews have more valuable information while long reviews contain more noises [13], we set the length of reviews in DeepCoNN is fixed to 100 words. Also, in NARRE we only include users and items with a minimum of 10 reviews each (details on DeepCoNN and NARRE are in Section V-A3).

3) Baselines

In this paper, we apply three non-review aware factorisation based recommender systems (NMF, SVD, SVD++), two state-of-the-art review-aware recommender systems (DeepCoNN and NARRE), and one novel text generation model (GCN) [12] to measure the performance of our methods. The details of these baselines are as follows:

- **NMF** [48]: Non-negative matrix factorization: a basic factorization method estimating two low-rank matrices for rating prediction.
- **SVD** [49]: Singular value decomposition: a popular collaborative filtering method that learns the relationship between users and latent factors.
- **SVD++** [50]: extends the SVD algorithm and incorporates implicit information.

[4]https://www.tensorflow.org

Review aware	Video		Toys		Beer		Yelp	
Recommender systems	DeepCoNN	NARRE	DeepCoNN	NARRE	DeepCoNN	NARRE	DeepCoNN	NARRE
human	0.898	0.898	0.878	0.878	1.175	1.157	0.861	0.859
GCN	0.897	0.888	0.878	0.875	1.174	**1.153**	0.861	0.858
Context	0.888	0.891	0.852	0.876	1.173	1.156	0.860	0.851
Attention	**0.881****	**0.867****	**0.845****	**0.874****	**1.156****	1.154	**0.852***	**0.850***

TABLE 4: RMSE Performance for review aware recommender systems. Note that human and GCN are the baselines, while Context and Attention are our models. * and ** denote the statistical significance for $p < 0.05$ and $p < 0.01$ respectively.

- **DeepCoNN** [3]: An Convolutional Neural Network based recommender system, in which the review text of users and items are modeled jointly.
- **NARRE** [5]: An enhanced version of DeepCoNN which uses an attention mechanism.
- **GCN** [12]: Generative Concatenative Network concatenates auxiliary information with sequential text as inputs to the recurrent network to generate personalised text.

Since this paper's goal is generating review-based explanations as input to review-aware recommender systems to improve recommendations, we take more concentration on the two review-aware recommender baselines: DeepCoNN and NARRE. To thoroughly leverage our achievements, we first compare our generated text with human-written text, then against with generated text from GCN.

4) Evaluation metrics

The primary target of this paper is to improve recommendation performance through machine-generated reviews. To this end, we employ RMSE to measure the rating prediction performance and NDCG@K to leverage the top-N ranking performance. Besides, We introduce the $\Delta RMSE$, which is defined in Equation. 10, to measure the recommendation improvements of machine-generated reviews over human-written reviews. Here, the greater the $\Delta RMSE$, the better performance for machine-generated reviews. For explanation evaluation, we describe evaluation methods and experiments in Section V-F.

- $\boldsymbol{RMSE}$ In line with [3], [5], [51], [52], we calculate Root Mean Square Error (RMSE) to evaluate rating predictions. It measures prediction errors of a group of user-item pairs. RMSE is defined in Equation. 9, where $\hat{R}_{u,i}$ represents predicted rating of user u on item i, while $R_{u,i}$ stands for their ground-truth rating. N is the total number of user-item pairs in *test* set.

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(\hat{R}_{u,i} - R_{u,i})^2} \tag{9}$$

- $\boldsymbol{\Delta RMSE}$ We introduce the $\Delta RMSE$, which is defined in Equation. 10, to measure the recommendation improvements of machine-generated reviews over human written reviews. Here, the greater the $\Delta RMSE$, the better performance for machine-generated reviews.

$$\Delta RMSE = RMSE_{human} - RMSE_{synthetic} \tag{10}$$

- **NDCG@K** NDCG@k [53]–[55] is a popular method to measure the effectiveness of predicted rankings, It evaluates the usefulness of items based on their ranking position. Higher NDCG@K values imply better item prediction order and this usually aligns better with the customers' interests.

B. RATING PREDICTION PERFORMANCE (Q1)

We follow the processes described in Figure 2 to execute our experiments. In this section, we evaluate the rating prediction performance by RMSE (Sec. V-A4). To perform a thorough analysis, we first compare the recommendations of our models with three non-review aware recommender systems (NMF, SVD, SVD++). Then we compare our model performance against human and automatic text generation model (GCN), using two review-aware recommender systems (DeepCoNN, NARRE). We demonstrate these comparisons in Table 3 and Table 4. Table 3 shows the rating prediction performance of non-review aware recommender systems, while Table 4 presents the recommendation outcomes on DeepCoNN and NARRE review-aware systems. According to these tables, we make several observations.

First, according to the RMSE in Table 3 and RMSE in the other Tables, it is clear that both human-written reviews and machine-generated reviews in combination with review-aware recommender systems show better performance than the systems which do not take into account the textual information. According to Chen *et al.* [5], human-written reviews can enhance the quality of the latent representation in recommender systems. Thus, we argue that generation models have learned the relevant patterns and contexts to improve the quality of their internal representations and out-perform the traditional non-review aware recommender systems.

Secondly, from Table 4, we can see that the machine-generated reviews consistently outperform human-written reviews on *all* datasets. This is a novel and somewhat surprising result. As we discussed previously, the quality of human-written reviews is variable, and we know that poorly written reviews can harm the prediction ability of the recommender systems. On the other hand, the RNNs we employed in our synthetic review generation models is an expert in learning

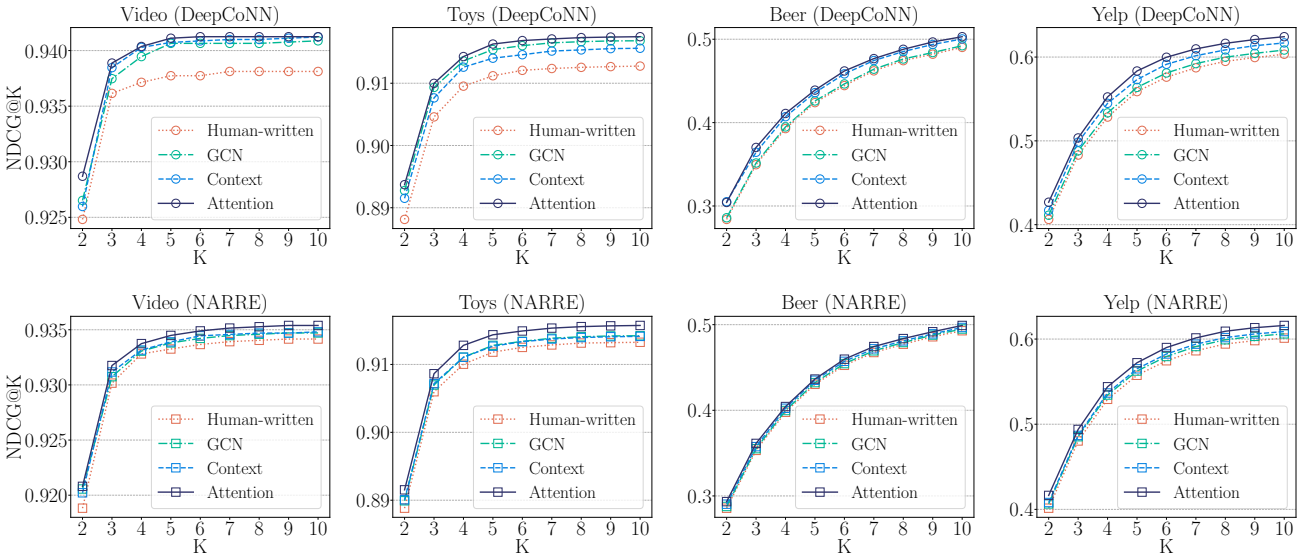


FIGURE 4: NDCG@K results on four datasets and two state-of-the-art text-aware recommender systems. Here, K ranges from 2 to 10. Test reviews are provided by human, GCN, Context and Attention models.

the most significant aspects of the text and reducing the generalisation error. Thereby, they can eliminate the impact of useless reviews and are trained to compose high-quality and relevant textual reviews.

Thirdly, comparing with GCN, the state-of-the-art generation model, both of our generation models exceed its performance in almost all cases, especially our Attention model. Although it has not surpassed GCN in the Beer dataset when using NARRE, the Attention model matches the performance of GCN, likely because the attention mechanism increased the ability of the model to learn more accurate syntax and better personalisation. Thus, we can generate more useful reviews and improve on the rating performances.

C. RANKING PERFORMANCE (Q1)

Another task of recommender systems is to recommend the correct items for users. In other words, the ideal recommendation is to rank items in a sequence that meets the users' preferences, which is a more onerous task than predicting ratings of items, and usually more valuable. In the ranking experiment, we start by producing K recommended items for each user in the test set through the two review aware recommender systems, DeepCoNN, and NARRE. As we described previously, we employ NDCG@K to leverage the recommended ranking quality, where higher NDCG@K value represents a more accurate ranking of items. Figure 4 reveals the ranking results. Firstly, we observe that the ranking performance of machine-generated reviews shows significant improvements over the ranking performance of human-written reviews in different levels of K. Though improvements in the Beer dataset are not as pronounced as in other datasets, we can still observe the continuous success that machine-generated reviews outperform human-written reviews in recommendation tasks. Moreover, our context model shows similar NDCG@K results to the GCN model, while our attention model surpasses GCN models on all datasets. It indicates that our attention model generates excellent reviews, leading to positive recommendation improvements.

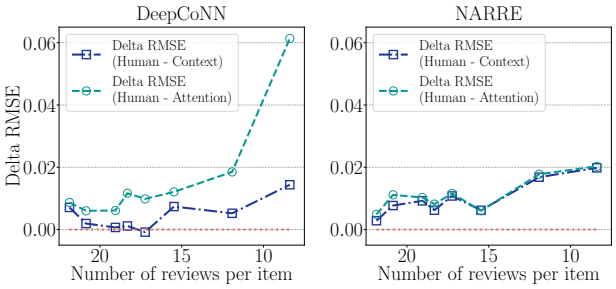


FIGURE 5: Recommendation performance comparison for different sparsity levels (number of reviews per item). We reduce the average number of reviews per item to increase the sparsity level.

D. PERFORMANCE W.R.T. SPARSITY (Q2)

As we discussed previously, review-aware recommender systems often suffer from poor performance when the rating information for users and items is sparse. Besides, low quality reviews harm the accuracy of recommendations, which is a more severe problem for review-aware recommender systems. On the other hand, text generation models learn rich representations for user-item pairs through the attribute

DeepCoNN	Video		Toys	
	A	B	A	B
Human	1.133**	0.929**	1.643**	1.025**
Context	1.100**	0.927**	1.421**	1.022**
Improvements	**2.89%**	**0.22%**	**13.53%**	**0.28%**

TABLE 5: Recommendation performance comparison for worthlessness group *A* (helpfulness ratio less or equal to 0.5) and usefulness group *B* (helpfulness ratio greater than 0.5). ** presents the statistical significance for $p < 0.01$.

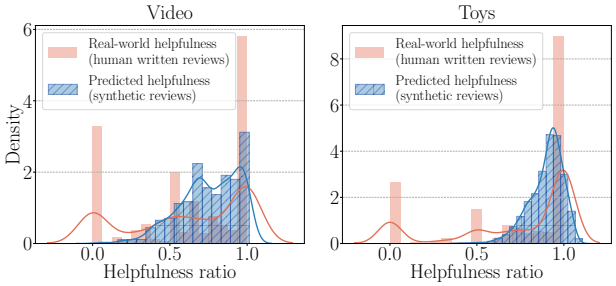


FIGURE 6: Distribution of helpfulness ratio of human-written reviews and helpfulness ratio of machine-generated reviews.

encoder module, introduced Section. IV. Thus, we argue that text generation models can learn significant knowledge from machine-generated reviews.

We design a set of experiments to measure the system performance in the presence of varying levels of sparsity. The most efficient way to create a sparse environment is removing records, as described in Feng *et al.* [56] who randomly removed ratings in the test data. In our experiment, we adopt the number of reviews per items to indicate the level of sparsity. Specifically, we remove reviews of items according to the distribution of the number of their reviews instead of removing them at random. Through this approach, we ensure all items have reviews and we can strictly regulate the level of sparsity.

Regarding the evaluation metric, we analyse the sparsity performance by considering the $\Delta RMSE$ for a variety of sparsity levels. We conduct our experiments on the Amazon Video dataset since it has the highest volume of reviews per item of all the datasets. Specifically, we manipulate the sparsity level of the Amazon Video dataset and rerun the experimental process (see Figure 2) for both human-written reviews and machine-generated reviews. We run this experiment on DeepCoNN and NARRE, two deep learning-based recommender systems, and show the results in Figure 5. By the definition of $\Delta RMSE$, greater $\Delta RMSE$ indicates the performance of generated machine-generated reviews surpasses human-written reviews. Experimental results demonstrate the effectiveness of generation models dealing with the sparsity problem. When the dataset becomes sparse, text generation models can still learn excellent user-item representations, which is how they achieve notable performance. In this way we address the sparsity problem for review-aware recommender systems.

E. PERFORMANCE W.R.T. HELPFUL REVIEWS (Q3)

Besides the sparsity issue, recommender systems also suffer from a text quality problem as we introduced in Section I. Some human written reviews contain tangential information and these *worthless reviews* serve to undermine recommendation quality. To improve user experience, many companies allow users to make subjective votes on whether the reviews from third-parties helped their decision (Amazon is a good example). In this experiment, we aim to explore whether generation models can improve the reviews that are voted as *not helpful*. We calculate the proportion of helpfulness votes as the *helpfulness ratio* for each review in two Amazon datasets. We then group the user-item pairs whose helpfulness ratio is greater than 0.5 into the usefulness group, otherwise the worthlessness group. We leverage the recommendation performance for both usefulness and worthlessness groups by RMSE. Besides, to assess the improvements outlined above, we calculate the percentage of the improvement for machine-generated reviews over human-written reviews. We conduct this experiment on DeepCoNN recommender system and demonstrate the results in Table 5. From these results, we can see that machine-generated reviews outperform human reviews in terms of both usefulness and worthlessness groups. Additionally, machine-generated reviews show significant $RMSE$ improvements for the worthlessness group. We assume that although generation models learn from human-written reviews, they concentrate on modelling useful instead of meaningless aspects that instead contribute to the noise and are ignored.

To validate the above assumption, we train an XGBoost model [57] and predict the helpfulness ratio for the machine-generated reviews in the test set. Figure 6 illustrates the distribution of real-word helpfulness ratio and predicted helpfulness ratio. According to these results, most reviews are helpful in the real world, while there is still a considerable fraction of useless reviews. When predicting the helpfulness ratio, only a few machine-generated reviews are predicted as unhelpful in Video dataset, and most machine-generated reviews in Toys dataset are seen as helpful. This finding validates the above hypothesis that generation models focus on learning *useful* instead of *worthless* aspects. Thereby, machine-generated reviews can achieve better recommendation performance than human-written reviews, which indicates they are suitable ingredients for text-aware recommender systems.

F. INTERPRETATION QUALITY AND SAMPLES (Q4)

The ideal way to evaluate the explanation performances is experimenting on real-world recommender systems, in which live users can conduct fair judgments. We plan to run a live-

Models	Perplexity				BLEU-4				TFIDF similarity				Readability similarity			
	Video	Toys	Beer	Yelp	Video	Toys	Beer	Yelp	Video	Toys	Beer	Yelp	Video	Toys	Beer	Yelp
N-gram	820.1	680.6	355.9	787.0	0.002	0.003	0.029	0.020	0.048	0.058	0.101	0.060	0.644	0.586	0.565	0.466
GCN	452.2	165.3	165.6	134.1	0.136	0.393	1.160	0.278	0.090	0.093	0.171	0.095	0.790	0.720	0.934	0.775
Context	194.4	164.0	159.2	108.6	0.403	0.676	3.409	0.395	0.109	0.122	0.182	0.117	0.782	0.739	0.945	0.866
Attention	**177.1**	**162.1**	**131.9**	**102.2**	**0.412**	**0.691**	**4.178**	**0.553**	**0.113**	**0.129**	**0.209**	**0.134**	**0.799**	**0.802**	**0.965**	**0.905**

TABLE 6: Explanation performance using the NLP and readability methods described in Sec. V-F. In all metrics our attention model shows the best performance.

user trial, but in this paper, we focus on first evaluating explanations by offline approaches, which can provide valuable information on the quality of the generated explanations. According to Hase *et al.* [58], many works use offline statistical methods to measure the quality of explanations quantitatively and conduct case studies for leveraging the performance of explanations qualitatively. Similarly, in this section, we first measure the quantitative performance of explanations on Natural Language Processing (NLP) methods. Then we leverage the qualitative performance of explanations by case studies.

For quantitative analyses, we employ four NLP evaluation methods: Perplexity [17], expressed as the exponentiation of the entropy per words, is a commonly used intrinsic methodology in natural language generation. It is used to measure how well the word probability distributions of the machine-generated reviews match those of the test reviews. Generally, lower perplexity means a better text generation. BLEU score [13], another well-known approach in machine translation and text generation tasks, measures the word correlations between machine-generated reviews and test reviews by calculating the precision of n-gram matching. TF-IDF similarity [17] is a statistical method reflecting the importance of words to review corpus. To leverage the relevance between machine-generated reviews and test reviews, we calculate the cosine similarity on TF-IDF of generated reviews and test reviews. Readability similarity [7] aims to evaluate whether a given review text is readable or not. We adopt eight readability algorithms and use the output readability value to represent reviews. The readability measures are: Automated Readability index [59], Flesch reading ease [60], Flesch-Kincaid grade level [60], Gunning-Fog index [60], simple measure of gobbledygook [61], Coleman Liau index [60], LIX [62] and RIX [62].

Notably, we compare our methods with two baselines, the classical N-gram language model [63], which predicts the occurrences of N consecutive words, and the GCN model. We run our experiments on the test set of four datasets, and the results are presented in Table 6. Through this table, we notice that our generation models consistently outperform the baselines. The context model beats the N-gram model and shows comparable performance to the GCN model, and the attention model shows the best explanation achievements. These promising results reflect our generated reviews are readable and adhere to the grammar and syntax of natural language expression.

We then conduct qualitative studies by empirically studying the explanation effect of machine-generated reviews, as shown in Table 7. From this table, we can observe that for the specific user, item, and rating, generation models can provide similar text quality to the human-written reviews. The first example demonstrates two reviews involving lego star wars, and they would like to recommend this item to others, which shows that our models can accurately reproduce the opinions and sentiments of a user about an item. In the second example, both machine-generated reviews and human-written reviews mention the price, which is one of the vital aspects that interests users. Moreover, the two reviews in the third example deliver a negative sentiment ("disappointed"), explaining why we do not recommend this item.

(User, Item)	Rating	Human written reviews	machine-generated reviews
(A, X)	**5**	Purchased as a christmas gift my year old grandson and he just loved them and could not wait until he built the lego star wars rebel trooper battle pack, highly recommend	this is the best thing that i can't say that the star wars this is a great toy for the price. I would recommend this to anyone who loves to play with lego
(B, Y)	**3**	I bought two sets i received small amount of energy that went with almost none of the pokemon and quiet a bit of trainers a few foils. A couple rares and some in japanese but not bad for the price	I don't know if it was a bit of a standard plastic product i would not recommend this product for any child, although the price is not bad .
(C, Z)	**1**	I bought this for my grandson he enjoyed it at first but the newness wore off. There are not a lot of colors that came with it. It was somewhat disappointing .	This was a big hit with my year old but i was a disappointed that the box is a bit of a bad toy. it is not a great toy for a young child to play with.

TABLE 7: Comparison of synthetic personalised reviews with human-written reviews. We generate machine-generated reviews for anonymous users and items on different ratings using the attention model. We highlight words that appear in both human-written and machine-generated reviews. The sentiment value of each word is highlighted with the polarity: positive is green , negative is red .

This observation shows that we can provide readable reviews that satisfy the requirements for explaining recommendation accurately to the user.

VI. CONCLUSIONS

In this paper, we have improved recommendation accuracy by inputting compelling machine-generated review-based explanations. Concretely, we developed two character-level review generation models and addressed both recommendation and explanation problems. We conducted experiments on four real-world datasets from different domains.

Experimental results revealed that our models consistently surpass the baselines and achieve state-of-the-art rating prediction performance. In ranking evaluation, our attention generation model outperformed baselines and showed excellent performance. Besides, we demonstrated the effectiveness of generation models in dealing with item sparsity problems. We also showed that generation models are suitable ingredients for recommender systems since they focus on generating useful machine-generated reviews.

To evaluate explanation quality, we used multiple offline metrics in the NLP field for quantitative evaluation and conduct case studies on generating review-based explanations in different combinations of users, items and ratings for qualitative evaluation. The quantitative evaluation results validated that our generated explanations received good scores for understandable. Also, our qualitative results demonstrate that we can produce critical words that express a user's real opinion of the item, where such critical words are the key aspects for delivering strong explainability.

These results as a whole are convincing arguments for the extensive use of machine-generated reviews to explain predicted ratings. Offline evaluation is not enough, our next steps will be continuing assessing the explanation quality through online live-user trials. In conclusion, improving the automatic generation of machine-generated reviews is a valuable next direction to provide more precise recommendations and personalised explanations.

REFERENCES

[1] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. E-commerce in your inbox: Product recommendations at scale. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1809–1818. ACM, 2015.

[2] Charu C Aggarwal et al. Recommender systems. Springer, 2016.

[3] Lei Zheng, Vahid Noroozi, and Philip S Yu. Joint deep modeling of users and items using reviews for recommendation. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, pages 425–434. ACM, 2017.

[4] Yuyun Chen, Hang Dong, and Wei Wang. Topic-graph based recommendation on social tagging systems: a study on research gate. In Proceedings of the 2018 International Conference on Data Science and Information Technology, pages 138–143. ACM, 2018.

[5] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Neural attentional rating regression with review-level explanations. In Proceedings of the 2018 World Wide Web Conference, pages 1583–1592, 2018.

[6] Bart P Knijnenburg, Martijn C Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. Explaining the user experience of recommender systems. User Modeling and User-Adapted Interaction, 22(4-5):441–504, 2012.

[7] Felipe Costa, Sixun Ouyang, Peter Dolog, and Aonghus Lawlor. Automatic generation of natural language explanations. In Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion, page 57. ACM, 2018.

[8] Mengting Wan and Julian McAuley. Item recommendation on monotonic behavior chains. In Proceedings of the 12th ACM Conference on Recommender Systems, pages 86–94. ACM, 2018.

[9] Yaqing Wang, Chunyan Feng, Caili Guo, Yunfei Chu, and Jenq-Neng Hwang. Solving the sparsity problem in recommendations via cross-domain item embedding based on co-clustering. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pages 717–725. ACM, 2019.

[10] Anindya Ghose and Panagiotis G Ipeirotis. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In Proceedings of the ninth international conference on Electronic commerce, pages 303–310, 2007.

[11] Nitin Jindal and Bing Liu. Opinion spam and analysis. In Proceedings of the 2008 international conference on web search and data mining, pages 219–230, 2008.

[12] Zachary C Lipton, Sharad Vikram, and Julian McAuley. Capturing meaning in product reviews with character-level generative text models. arXiv preprint arXiv:1511.03683, 2015.

[13] Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. Learning to generate product reviews from attributes. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, volume 1, pages 623–632, 2017.

[14] Guoshuai Zhao, Hao Fu, Ruihua Song, Tetsuya Sakai, Xing Xie, and Xueming Qian. Why you should listen to this song: Reason generation for explainable recommendation. In 2018 IEEE International Conference on Data Mining Workshops (ICDMW), pages 1316–1322. IEEE, 2018.

[15] Jianmo Ni, Zachary C Lipton, Sharad Vikram, and Julian McAuley. Estimating reactions and recommending products with generative models of reviews. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 783–791, 2017.

[16] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. Neural rating regression with abstractive tips generation for recommendation. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pages 345–354, 2017.

[17] Yichao Lu, Ruihai Dong, and Barry Smyth. Why i like it: multi-task learning for recommendation and explanation. In Proceedings of the 12th ACM Conference on Recommender Systems, pages 4–12. ACM, 2018.

[18] PVS Avinesh, Yongli Ren, Christian M Meyer, Jeffrey Chan, Zhifeng Bao, and Mark Sanderson. J3r: Joint multi-task learning of ratings and review summaries for explainable recommendation. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 339–355. Springer, 2019.

[19] Zhongxia Chen, Xiting Wang, Xing Xie, Tong Wu, Guoqing Bu, Yining Wang, and Enhong Chen. Co-attentive multi-task learning for explainable recommendation. In IJCAI, pages 2137–2143, 2019.

[20] Quoc-Tuan Truong and Hady Lauw. Multimodal review generation for recommender systems. In The World Wide Web Conference, pages 1864–1874, 2019.

[21] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten De Rijke. Explainable outfit recommendation with joint outfit matching and comment generation. IEEE Transactions on Knowledge and Data Engineering, 2019.

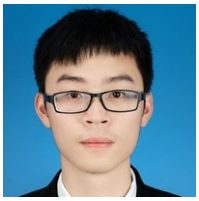
[22] Octavio Loyola-Gonzalez. Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view. IEEE Access, 7:154096–154113, 2019.

[23] Raymond J Mooney, Paul N Bennett, and Loriene Roy. Book recommending using text categorization with extracted information. In Proc. Recommender Systems Papers from 1998 Workshop, Technical Report WS-98-08, 1998.

[24] Robin Van Meteren and Maarten Van Someren. Using content-based filtering for recommendation. In Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop, pages 47–56, 2000.

[25] Amjad Almahairi, Kyle Kastner, Kyunghyun Cho, and Aaron Courville. Learning distributed representations from reviews for collaborative filtering. In Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, pages 147–154, New York, NY, USA, 2015. ACM.

[26] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In Proceedings of the Eleventh ACM Conference on Recommender Systems, pages 297–305, 2017.
[27] Ting Chen, Liangjie Hong, Yue Shi, and Yizhou Sun. Joint text embedding for personalized content-based recommendation. arXiv preprint arXiv:1706.01084, 2017.
[28] Emanuel Lacic, Dominik Kowald, and Elisabeth Lex. High enough?: Explaining and predicting traveler satisfaction using airline reviews. In Proceedings of the 27th ACM Conference on Hypertext and Social Media, pages 249–254. ACM, 2016.
[29] Shuo Chang, F Maxwell Harper, and Loren Gilbert Terveen. Crowd-based personalized natural language explanations for recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, pages 175–182. ACM, 2016.
[30] Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Explod: a framework for explaining recommendations based on the linked open data cloud. In Proceedings of the 10th ACM Conference on Recommender Systems, pages 151–154. ACM, 2016.
[31] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.
[32] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. Social collaborative viewpoint regression with explainable recommendations. In Proceedings of the tenth ACM international conference on web search and data mining, pages 485–494, 2017.
[33] Lu Wang, Wei Zhang, Xiaofeng He, and Hongyuan Zha. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2447–2456. ACM, 2018.
[34] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pages 1017–1024, 2011.
[35] Jian Tang, Yifan Yang, Sam Carton, Ming Zhang, and Qiaozhu Mei. Context-aware natural language generation with recurrent neural networks. arXiv preprint arXiv:1611.09900, 2016.
[36] Chen Ma, Peng Kang, Bin Wu, Qinglong Wang, and Xue Liu. Gated attentive-autoencoder for content-aware recommendation. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pages 519–527, 2019.
[37] Apurva Pathak, Kshitiz Gupta, and Julian McAuley. Generating and personalizing bundle recommendations on steam. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 1073–1076. ACM, 2017.
[38] Hercules Dalianis and Eduard Hovy. Aggregation in natural language generation. In European Workshop on Trends in Natural Language Generation, pages 88–105. Springer, 1993.
[39] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
[40] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
[41] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 4806–4813, 2019.
[42] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 43–52. ACM, 2015.
[43] Julian John McAuley and Jure Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In Proceedings of the 22nd international conference on World Wide Web, pages 897–908. ACM, 2013.
[44] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078, 2015.
[45] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[46] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1):1929–1958, 2014.
[47] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In International Conference on Machine Learning, pages 1310–1318, 2013.
[48] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In Advances in neural information processing systems, pages 556–562, 2001.
[49] Gilbert Strang. The fundamental theorem of linear algebra. The American Mathematical Monthly, 100(9):848–855, 1993.
[50] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 426–434. ACM, 2008.
[51] Jiliang Tang, Suhang Wang, Xia Hu, Dawei Yin, Yingzhou Bi, Yi Chang, and Huan Liu. Recommendation with social dimensions. In Thirtieth AAAI Conference on Artificial Intelligence, 2016.
[52] Jianling Wang and James Caverlee. Recurrent recommendation with local coherence. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pages 564–572. ACM, 2019.
[53] Ziwei Zhu, Jianling Wang, and James Caverlee. Improving top-k recommendation via jointcollaborative autoencoders. In The World Wide Web Conference, pages 3483–3482. ACM, 2019.
[54] Athanasios N Nikolakopoulos and George Karypis. Recwalk: Nearly uncoupled random walks for top-n recommendation. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pages 150–158. ACM, 2019.
[55] Zhi-Hong Deng, Ling Huang, Chang-Dong Wang, Jian-Huang Lai, and Philip S Yu. Deepcf: A unified framework of representation learning and matching function learning in recommender system. arXiv preprint arXiv:1901.04704, 2019.
[56] Chenjiao Feng, Jiye Liang, Peng Song, and Zhiqiang Wang. A fusion collaborative filtering method for sparse data in recommender systems. Information Sciences, 521:365–379, 2020.
[57] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pages 785–794. ACM, 2016.
[58] Peter Hase and Mohit Bansal. Evaluating explainable ai: Which algorithmic explanations help users predict model behavior? arXiv preprint arXiv:2005.01831, 2020.
[59] Lei Liu, Georgia Koutrika, and Shanchan Wu. Learningassistant: A novel learning resource recommendation system. In Data Engineering (ICDE), 2015 IEEE 31st International Conference on, pages 1424–1427. IEEE, 2015.
[60] Maria Soledad Pera and Yiu-Kai Ng. What to read next?: Making personalized book recommendations for k-12 users. In Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, pages 113–120, New York, NY, USA, 2013. ACM.
[61] G Harry Mc Laughlin. Smog grading-a new readability formula. Journal of reading, 12(8):639–646, 1969.
[62] Jonathan Anderson. Lix and rix: Variations on a little-known readability index. Journal of Reading, 26(6):490–496, 1983.
[63] Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In 1995 International Conference on Acoustics, Speech, and Signal Processing, volume 1, pages 181–184. IEEE, 1995.



SIXUN OUYANG received a Master degree in Computer Science from University College Dublin (UCD), Ireland. Sixun is currently an independent researcher at the Insight Centre for Data Analytics, and pursuing the PhD degree in Computer Science at University College Dublin (UCD), Ireland. Sixun has rich experience in the fields of Machine Learning, Deep Learning, Explainable Systems, Natural Language Processing and Recommender Systems. Sixun has been enthusiastically engaged in large industrial projects and Europe National research projects. Sixun has received funded grants from the Science Foundation of Ireland.

AONGHUS LAWLOR is an Assistant Professor at the University College Dublin (UCD) and a Principal Investigator at the Insight Centre for Data Analytics. Aonghus holds a PhD from UCD and a Master of Science from the University of Cambridge. Aonghus has long experience in the areas of Machine Learning, Artificial Intelligence, Explainable AI, Natural Language Processing and Recommender Systems. Aonghus has been actively involved in several EU and National research projects in the above mentioned areas and has been the co-PI of very large Industrial projects. Aonghus has also received several funded grants from Enterprise Ireland and the Science Foundation of Ireland, leading to multiple patent applications.