Toward Explainable Recommendations: Generating Review Text from Multicriteria Evaluation Data

Takafumi Suzuki
Hokkaido University
suzuki_tk4@complex.ist.hokudai.ac.jp

Satoshi Oyama
Hokkaido University/RIKEN AIP
oyama@ist.hokudai.ac.jp

Masahito Kurihara
Hokkaido University
kurihara@ist.hokudai.ac.jp

***Abstract*—Explaining recommendations helps users to make more accurate and effective decisions and improves system credibility and transparency. Current explainable recommender systems tend to provide fixed statements such as "customers who purchased this item also purchased...". This explanation is generated only on the basis of the purchase history of similar customers, so it does not include the preferences of customers who have purchased the item or a description of the item. Since user-generated reviews generally contain information about the reviewer's preferences and a description of the item, such reviews typically have more effect on purchase decisions. Therefore, using reviews to explain recommendations should be more useful than providing only a fixed statement explanation. Aiming to create a system that provides personalized explanations for recommendations, we have developed a recurrent neural network model that uses multicriteria evaluation data to generate reviews.**

***Index Terms*—explainable recommendation, text generation, RNN, recommender systems**

I. INTRODUCTION

Though many collaborative filtering algorithms, which predict item ratings using purchasing histories, for recommender systems have been proposed, recent state-of-the-art methods are so complex that they lack interpretability, and the recommendation process is a 'black box' to users. Improving the explainability of recommender systems is thus important. Explaining recommendations helps users to make more accurate and effective decisions, so that mismatches between users and items are decreased, and purchase behavior is promoted. As a result, the credibility and transparency of the recommendation system are increased. Current explainable recommender systems tend to offer fixed statements such as "customers who purchased this item also purchased ... ". This explanation is generated only on the basis of the purchase history of customers similar to the user in terms of purchase behavior. It therefore does not include preferences of customers who have purchased the item or a description of the item. Since user-generated reviews generally contain information about the reviewer's preferences and a description of the item, such reviews typically have more effect on purchase decisions. Therefore, using reviews to explain recommendations should be more useful than providing only a fixed statement explanation. For more personalized explanations, we have taken an approach as follows: synthesize review text from multicriteria evaluation data containing more detailed information useful for personalized recommendation [1].

There have been many studies on deep learning, especially the use of recurrent neural networks (RNNs) for text generation. Several studies have presented RNN models for generating review texts [2] [3]. We have extended the model proposed by Dong et al. [3] for generating review texts that include user preferences and item features. To enable the use of information about users and items for generating reviews, we use implicit feedback. We used the TripAdvisor dataset [4], which is a multicriteria evaluation dataset.

II. RELATED WORK

Explainability is important in many respects, e.g. increasing system *transparency*, system *persuasiveness*, user *trust*, user *satisfaction*, users' *decision efficiency*, and users' *decision effectiveness* [5]. Some explanation interfaces use a rating-based approach. For example, a system might group users on the basis of the ratings they have given and display a histogram of "neighbor users"as explanation [6]. Another approach for example is that taken by Chang et al. [7]: prepare natural language explanations using crowdsourcing.

Ni et al. [2] proposed several models for generating reviews by using character-level RNNs, in which the character used to generate text is recurrently predicted. Several attention mechanisms for generating review text were devised by Dong et al. [3] that aim to better utilize the input information. The model proposed by Dong et al. considers only single-criteria evaluation. We have extended the model so that it can handle multicriteria evaluation data and evaluated its ability to capture user preferences and item features.

III. EXTENDED MODEL

Our goal is a model that, when given attributes $a = (a_1, a_2, ..., a_{|a|})$, can generate a product review $r = (y_1, y_2, ..., r_{|r|})$ maximizing the likelihood of generated review texts given input attributes $p(r \mid a)$. The number of attributes $\mid a \mid$ is fixed, and the review length is variable.

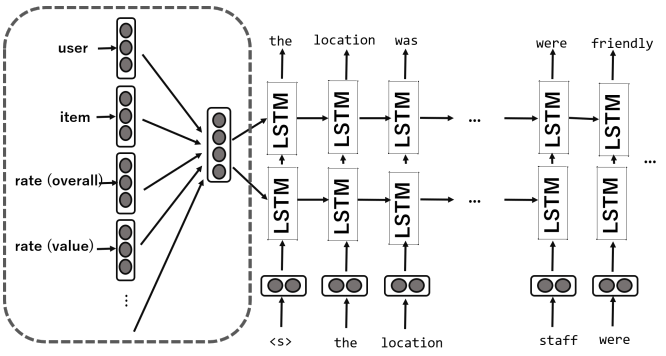

Fig. 1. Extended model without attention mechanism

We use userID, itemID, overall rating, and multicriteria evaluation from the TripAdvisor dataset as attributes. The training data is a pair of attributes and the corresponding reviews. The model computes likelihood $p(r \mid a)$. The likelihood is decomposed into,

$$p(r \mid a) = \prod_{t=1}^{|r|} p(y_t \mid y_{<t}, a), \tag{1}$$

where $y_{<t} = (y_1, y_2, ..., y_{t-1})$.

Our extended model is composed of an attribute encoder, a sequence decoder, and an attention layer. We use a multilayer perceptron as the attributes encoder. It encodes the attributes into context vectors. The sequence decoder uses long short-term memory (LSTM) units [8] to decode the context vectors into reviews by predicting the next word in each time step. An attention mechanism is used to better utilize the attributes. The attention layer uses the attributes vector to predict the next word. The extended model without the attention mechanism is illustrated in Fig.1.

A. Encoder

When encoding attributes using a multilayer perceptron, each attribute vector $\mathbf{g}(a_i)$ is computed using $\mathbf{g}(a_i) = W_i^a \mathbf{e}(a_i)$ where $W_i^a \in \mathbb{R}^{m \times |a_i|}$ is a parameter matrix, m is the embedding dimension, and $\mathbf{e}(a_i) \in \{0,1\}^{|a_i|}$ is a one-hot vector. These attribute vector are concatenated and then passed to the hidden layers,$\mathbf{a} = \tanh(H[\mathbf{g}(a_1), ..., \mathbf{g}(a_{|a|})] + \mathbf{b}_a)$,

where $[\mathbf{g}(a_1), ..., \mathbf{g}(a_{|a|})]$ is the concatenated vector, $H \in \mathbb{R}^{L_n \times |a| m}$ is the weight matrix, and $\mathbf{b}_a \in \mathbb{R}^{L_n}$ are the bias parameters. Vector $\mathbf{a}$ is used to initialize the decoder's L hidden layers.

B. Decoder

The RNN with LSTM units [8] used as the sequence decoder uses the hidden layer to represent the current information and recurrently computes the next hidden layer using the current hidden state. The lth hidden layer for time step t $\mathbf{h}_t \in \mathbb{R}^n$ is computed as follows $\mathbf{h}_t^l = f(\mathbf{h}_{t-1}^l, \mathbf{h}_t^{l-1})$, where f is a function denoting the recurrent computation of the RNN for time step t, $\mathbf{h}_0^l = W^r \mathbf{e}(y_{t-1})$ is the word embedding of the previous predicted word, $W^r \in \mathbb{R}^{n \times |V_r|}$ is a parameter matrix, and $\mid V_r \mid$ is the vocabulary size. The input attributes are encoded into a vector $\mathbf{a}$ and then used to initialize the hidden layers for the first time step: $\mathbf{h}_0^l = f(\mathbf{a}, \mathbf{h}_0^{l-1})$. Then, the decoder predicts the output words using the topmost layer of the LSTM units. For the model without the attention mechanism, the probability distribution for the output words is $p(y_t \mid y_{<t}, a) = \text{softmax}(W^p \mathbf{h}_t^L)$, where $W^p \in \mathbb{R}^{|V_r| \times n}$ is a parameter matrix.

C. Attention mechanism

The attention mechanism helps the decoder concentrate on the different parts of the encoded information. When the decoder generates sequences, the attention weights are computed using $\mathbf{g}(a_i)$ and the context vector $\mathbf{c}$ is computed by weighted sum of $\mathbf{g}(a_i)$. The decoder predicts the word at next time step as:

$$\mathbf{h}_t^{att} = \tanh(W_1 \mathbf{c}^t + W_2 \mathbf{h}_t^L) \tag{2}$$

$$p(y_t \mid y_{<t}, a) = \text{softmax}_{y_t}(W^p \mathbf{h}_t^{att}), \tag{3}$$

where $W^p \in \mathbb{R}^{|V_r| \times n}, W_1 \in \mathbb{R}^{n \times m}$, and $W_2 \in \mathbb{R}^{n \times n}$ is a parameter matrix.

D. Model Training

The model is trained to maximize the likelihood of generated review texts given input attributes for the training data. The RMSProp optimizer is used in the same way as the existing model [3] to optimize the objective function.

E. Inference

At test time, the trained model encodes the attributes in the test data set and decodes the context vector into review texts by maximizing the conditional probability defined in Equation (1). A greedy search algorithm is used when the decoder generates reviews. Specifically, the decoder predicts a word that maximizes Equation (1) at each time step. This avoids iterating over all candidate reviews. When the decoder predicts the end-of-sequence token, decoding is terminated.

IV. EVALUATION

The model was trained using a data set containing customer reviews. It then encoded the ID and attributes of each review and generated review texts. We evaluated the generated texts qualitatively by comparing their contents with those of the original review texts, focusing on certain aspects.

A. Data Description

The dataset contained TripAdvisor user review data [4]: userID, hotelID, overall rating, multicriteria rating (*value, room, location, cleanliness, check-in/front desk, service, business service*) and review. The ratings were 0–5.

B. Setting

The dimension of the attribute vectors was 64, and that of word embedding was 512. We used two-layer LSTM units as the decoder. The parameters of the encoder and decoder were randomly initialized by sampling from a uniform distribution $[-0.08, 0.08]$. The batch size, smoothing constant, and base learning rate of the RMSProp optimizer were set to 50, 0.95, and 0.002. After ten epochs, the learning rate was reduced by a factor of 0.97. We used Dropout as a regularizer, and the probability of an element being set to zero was 0.2. The gradients were clipped to a range of $[-5, 5]$.

C. Examples of generated review texts

Examples of the generated (top) and original (bottom) review texts and the corresponding rating (overall, value, room, location, cleanliness, check-in/front desk, service, business service) are shown in TableI. Some data were omitted due to space limitations. A rating of -1 means that that aspect was not evaluated. As evident in the table, the contents of the generated texts mostly corresponded to those of the original texts, with the same words often used in both texts. In the first example, each aspect expressed in the multicriteria evaluation is similarly mentioned, while in the second example, aspects are mentioned in the generated text that are not mentioned in the original text. This indicates that the model roughly learns user preferences and item features but does not learn detailed aspects, especially in the case of a low rating.

TABLE I
TWO EXAMPLES OF GENERATED (TOP) AND ORIGINAL (BOTTOM) REVIEW TEXTS

ratings	review texts
4, 5, 4, -1, 5, -1, 4, -1	good size and very clean ! the hotel was very clean and the staff were very friendly . we would recommend this hotel and would stay there again .
	good size and very clean definately recommend ! the hotel was round the corner from la motte picquet grenelle metro so very convenient . the staff spoke good english and were helpful . the lift was small but could (num) people with travel cases in .
1, 1, 1, 2, 3, 1, 1, 3	don't stay here ! we stayed there for (num) nights in may , and it was a disappointment . it was a disappointment . we had booked a deluxe room , which was a mistake . it was a tiny room , and the bathroom was tiny , with a sink that had a sink with a sink .
	don't stay here ! ... however , is just old , musty and lacks charm . further , the hosts were quite rude . there is no elevator/lift and the only access is via a narrow read one-way staircase consisting of (num)(num)steps with a turn .

V. CONCLUSION

Our extension of Dong et al.'s model generates review texts using multicriteria evaluation data. Our aim is a system that produces more personalized explanations. The experimental evaluation revealed that the quality of the generated review texts needs to be improved as they sometimes did not mention certain aspects or unnecessarily mentioned certain aspects. This is attributed to the large number of attributes with respect to the amount of data, the great variation in the lengths of the reviews, and the evaluation value is largely due to sparse.

Future work includes assessing whether the review generation model is valid as a recommendation explanation. Human evaluation by crowdsourcing is one way to do this. We could ask crowd workers to check whether generated texts mention each aspect mentioned in the user reviews. Another way to do this is to use a recommendation algorithm. If we had a reliable recommendation algorithm that uses review texts, we could input generated and original review texts and compare the outputs to evaluate the generated texts [9]. We could also analyze whether the references to aspects in the generated review text matches the user's preferences by using an aspect sentiment analysis technique [4]. Given that various evaluation methods are possible, we will survey while using algorithms and human evaluation together for the effectiveness and validity of the recommendation explanation.

ACKNOWLEDGEMENTS

This work was partially supported by JSPS KAKENHI Grant Number JP18H03337.

REFERENCES

[1] H. Morise, S. Oyama, and M. Kurihara, "Collaborative filtering and rating aggregation based on multicriteria rating," in *HM-Data*, 2017.

[2] J. Ni, Z. C. Lipton, S. Vikram, and J. McAuley, "Estimating reactions and recommending products with generative models of reviews," in *IJNLP*, 2017.

[3] L. Dong, S. Huang, F. Wei, M. Lapata, M. Zhou, and K. Xu, "Learning to generate product reviews from attributes," in *EACL*, 2017.

[4] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis on review text data: a rating regression approach," in *KDD*, 2010.

[5] N. Tintarev and J. Masthoff, "Evaluating the effectiveness of explanations for recommender systems," *User Modeling and User-Adapted Interaction*, vol. 22, no. 4-5, 2012.

[6] J. L. Herlocker, J. A. Konstan, and J. Riedl, "Explaining collaborative filtering recommendations," in *CSCW*, 2000.

[7] S. Chang, F. M. Harper, and L. G. Terveen, "Crowd-based personalized natural language explanations for recommendations," in *RecSys*, 2016.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, 1997.

[9] S. Ouyang, A. Lawlor, F. Costa, and P. Dolog, "Improving explainable recommendations with synthetic reviews," *arXiv preprint arXiv:1807.06978*, 2018.