BERT-Based Neural Collaborative Filtering and Fixed-Length Contiguous Tokens Explanation

Reinald Adrian Pugoy[1,2] and **Hung-Yu Kao**[1]
[1]National Cheng Kung University, Tainan City, Taiwan
[2]University of the Philippines Open University, Los Baños, Philippines
rdpugoy@up.edu.ph, hykao@mail.ncku.edu.tw

Abstract

We propose a novel, accurate, and explainable recommender model (BENEFICT) that addresses two drawbacks that most review-based recommender systems face. First is their utilization of traditional word embeddings that could influence prediction performance due to their inability to model the word semantics' dynamic characteristic. Second is their black-box nature that makes the explanations behind every prediction obscure. Our model uniquely integrates three key elements: BERT, multilayer perceptron, and maximum subarray problem to derive contextualized review features, model user-item interactions, and generate explanations, respectively. Our experiments show that BENEFICT consistently outperforms other state-of-the-art models by an average improvement gain of nearly 7%. Based on the human judges' assessment, the BENEFICT-produced explanations can capture the essence of the customer's preference and help future customers make purchasing decisions. To the best of our knowledge, our model is one of the first recommender models to utilize BERT for neural collaborative filtering.

1 Introduction

In recommender systems research, collaborative filtering (CF) is the dominant state-of-the-art recommendation model, which primarily focuses on learning accurate representations of users (user preferences) and items (item characteristics) (Chen et al., 2018; Tay et al., 2018). The earliest recommender models learned these representations based on user-given numeric ratings that each item received (Mnih and Salakhutdinov, 2008; Koren et al., 2009). However, ratings, which are values on a single discrete scale, oversimplify user preferences and item characteristics (Musto et al., 2017). The large amount of users and items in a typical online platform consequently results in a highly sparse rating matrix, making it hard to learn accurate representations (Zheng et al., 2017).

To alleviate these issues, review texts have instead been utilized to model such representations for subsequent recommendation and rating prediction, and this approach has attracted growing attention in research (Catherine and Cohen, 2017; Zheng et al., 2017). The main advantage of reviews as the source of features is that they can cover user opinions' multi-faceted substance. Because users can explain their reasons underlying their given ratings, reviews contain a large amount of latent information that is both rich and valuable, and that cannot be otherwise obtained from ratings alone (Chen et al., 2018; Wang et al., 2019). Recently, models that incorporate user reviews have yielded state-of-the-art performances (Zheng et al., 2017; Chen et al., 2018). These approaches learn user and item representations by using traditional word embeddings (e.g., word2vec, GloVe) to map each word in the review into its corresponding vector. The review is transformed into an embedded matrix before being fed to a convolutional neural network (CNN) (Chen et al., 2018). CNNs have been shown to effectively model reviews and have illustrated outstanding results in numerous natural language processing tasks (Wang et al., 2018a).

Nevertheless, there are drawbacks that most review-based recommender models experience. First is the utilization of traditional or mainstream word embeddings to learn review features. Their static nature is a hindrance, as each word sense is associated with the same embedding regardless of the context. In other words, such embeddings cannot identify the dynamic nature of each word's semantics. For review-based recommenders, this could be an issue in modeling users and items, which could, in turn, affect recommendation performance (Pilehvar and Camacho-Collados, 2019). Also, once a CNN is fed with the matrix of word embeddings, the word frequency information of contextual fea-

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 143–153
December 4 - 7, 2020. ©2020 Association for Computational Linguistics

tures, said to be crucial for modeling reviews, will be lost (Wang et al., 2018a).

Another drawback is the inherent black-box nature of deep learning-based models that makes the explanations behind every prediction obscure (Ribeiro et al., 2016; Wang et al., 2018b). The complex architecture of hidden layers has opaqued the models' internal decision-making processes (Peake and Wang, 2018). Providing explanations could help persuade users to make decisions and develop trust in a recommender system (Zhang et al., 2014; Ribeiro et al., 2016; Costa et al., 2018; Peake and Wang, 2018). However, this leads us to a dilemma, i.e., a trade-off between accuracy and explainability. Usually, the most accurate models are inherently complicated, non-transparent, and unexplainable (Zhang and Chen, 2018). The same is also true for explainable and straightforward methods that sacrifice accuracy. Formulating models that are both explainable and accurate is a challenging yet critical research agenda for the machine learning community to ensure that we derive benefits from machine learning fairly and responsibly (Peake and Wang, 2018).

In this paper, we propose a unique model: **BE**RT-Based **Ne**ural Collaborative Filtering and **Fi**xed-Length **C**ontiguous **T**okens Explanation **(BENEFICT)**. Our model learns user and item representations simultaneously using two parallel networks. To address the first drawback, we incorporate BERT as a key component in each parallel network. BERT affords us to extract more meaningful, contextualized features adaptable to arbitrary contexts; such features cannot be derived from mainstream word embeddings (Pilehvar and Camacho-Collados, 2019; Zakbik et al., 2019). BERT can also retain the word frequency information that makes CNN an unnecessary component of our model. Once user and item representations are learned, they are concatenated together in a shared hidden space before being finally fed to an optimal stack of multilayer perceptron (MLP) layers that serve as BENEFICT's interaction function.

To address the second drawback, we introduce a novel component in our model that integrates BERT's self-attention and an implementation of the fixed-length maximum subarray problem (MSP), which is considered to be a classic computer science problem. BERT applies self-attention in each encoder layer that consequently produces self-attention weights for each token. These are passed to the successive encoder layers through feedforward networks. We argue that these self-attention weights can be the basis for explaining rating predictions. Based on this premise, MSP then selects a segment or subarray of consecutive tokens that has the maximum possible sum of self-attention weights.

1.1 Contributions

Our work aims to fill the research gap by implementing a solution that is both accurate and explainable. We propose a novel model that uniquely integrates three vital elements, i.e., BERT, MLP, and MSP, to derive review features, model user-item interactions, and produce possible explanations. To the best of our knowledge, BENEFICT is one of the first review-based recommender models to utilize BERT for neural CF. Also, to the extent of our knowledge, BENEFICT is one of the first models to repurpose a portion of the Neural Collaborative Filtering (NCF) framework (He et al., 2017) as the user-item interaction function for review-based, explicit CF. Moreover, our experiments have demonstrated that our model achieves better rating prediction results than the other state-of-the-art recommender models.

2 Related Work and Concepts

Designing a CF model involves two crucial steps: learning user and item representations and modeling user-item interactions based on those representations (He et al., 2018). Before the advancements provided by neural networks, matrix factorization (MF) was the dominant model representing users and items as vectors of latent factors (called embeddings) and models user-item interactions using the inner product operation. The said operation leads to poor performance because it is sub-optimal for learning rich yet complicated patterns from real-world data (He et al., 2018). To address this scenario, neural networks (NN) have been integrated into recommender architectures. One of the initial works that have laid the foundation in employing NN for CF is NCF (He et al., 2017). Their framework, originally implemented for rating-based, implicit CF, learns non-linear interactions between users and items by employing MLP layers as their interaction function, granting it a high degree of non-linearity and flexibility to learn meaningful interactions. Two common designs have emerged when it comes to leveraging MLP layers: placing an MLP above either the concatenated user-item embeddings (He et al., 2017; Bai et al., 2017) or the element-wise product of user and item embeddings (Zhang et al., 2017; Wang et al., 2017).

As far as rating prediction is concerned, two notable recommender models have yielded significant state-of-the-art prediction performances. DeepCoNN is the first deep model that represents users and items from reviews jointly (Zheng et al., 2017). It consists of two parallel, CNN-powered networks. One network learns user behavior by examining all reviews that he has written, and the other network models item properties by exploring all reviews that it has received. A shared layer connects these two networks, and factorization machines capture user-item interactions. The second model is NARRE, which shares certain similarities with DeepCoNN. NARRE is also composed of two parallel networks for user and item modeling with respective CNNs to process reviews (Chen et al., 2018). Rather than concatenating reviews to one long sequence the same way that DeepCoNN does, their model introduces an attention mechanism that learns review-level usefulness in the form of attention weights. These weights are integrated into user and item representations to enhance the embedding quality and the subsequent prediction accuracy. Both DeepCoNN and NARRE employ traditional word embeddings.

Other relevant studies have claimed to provide explanations for recommendations such as EFM (Zhang et al., 2014), sCVR (Ren et al., 2017), and TriRank (He et al., 2015). These models initially extract aspects and opinions by performing phrase-level sentiment analysis on reviews. Afterward, they generate feature-level explanations according to product features that correspond to user interests (Chen et al., 2018). However, these models have some limitations; manual preprocessing is required for sentiment analysis and feature extraction, and the explanations are simple extraction of words or phrases from the review text (Zhang et al., 2014; Ren et al., 2017). This also has the unintended effect of distorting the reviews' original meaning (Ribeiro et al., 2016; Chen et al., 2018). Another limitation is that textual similarity is solely based on lexical similarity; this implies that semantic meaning is ignored (Zheng et al., 2017; Chen et al., 2018).

3 Methodology

BENEFICT, as illustrated in Figure 1, has two parallel networks to model user and item embeddings that both utilize BERT. Hereafter, we will only illustrate the user modeling process because the same is also observed for item modeling, with their inputs as the only difference.

3.1 Input Layer and BERT Encoding

Given an input set of user-written reviews $V_u = \{V_{u1}, V_{u2}, ..., V_{uj}\}$ where j is the total number of reviews from user u, V_u is fed to a pre-trained BERT$_{\text{BASE}}$ model to encode the reviews and obtain their respective contextualized representations. BERT$_{\text{BASE}}$ consists of 12 encoder layers and 12 self-attention heads (Devlin et al., 2018). It also has a hidden size of 768, which we will directly utilize later as the fixed embedding dimension. Furthermore, BERT requires every review to follow a particular format. For this purpose, the model applies WordPiece tokenization to the review's input sequence (Wu et al., 2016). The format is comprised of token embeddings, segment embeddings, position embeddings, and padding masks. Because rating prediction is not a sentence pairing task, BERT takes each review as a single segment of contiguous text. Typically, BERT supports a maximum sequence length of 512 tokens. In this study, we use a shorter length of 256 tokens to save substantial memory. As such, each input sequence is truncated or padded accordingly.

The newly-formatted input sequence then passes through a stack of Transformer encoders to obtain the contextualized representations of reviews: $h_{\texttt{[CLS]},u} = \{h_{\texttt{[CLS]},u1}, h_{\texttt{[CLS]},u2}, ..., h_{\texttt{[CLS]},uj}\}$, where $h_{\texttt{[CLS]},u} \in \mathbb{R}^{j\times 768}$. We utilize the hidden state of the special `[CLS]` token to serve as the review's aggregate sequence representation or pooled contextualized embedding (Devlin et al., 2018). In theory, any encoder layer may be selected to provide the hidden state of `[CLS]` as the review's representation. We select the twelfth layer for our approach; prior studies have illustrated that its predictive capability is the best among the other layers (Sun et al., 2019).

3.2 Embedding Generation, Multilayer Perceptron, and Prediction

The user embedding (user feature vector) $P_u \in \mathbb{R}^{1\times 768}$ is obtained by calculating the average of the `[CLS]` representations of the reviews written by user u, given by the formula below. Similarly, the item embedding (item feature vector) $Q_i \in \mathbb{R}^{1\times768}$ can be generated from the item modeling network.

$$P_u = \frac{1}{j}\sum_{t=1}^{j} h_{\texttt{[CLS]},ut} \tag{1}$$

Furthermore, the purpose of incorporating an MLP is to learn the interactions between user and item representations and to model the CF effect, which will not be properly covered by solely using vector concatenation or element-wise product (He et al., 2017). Adding a certain number of hidden layers on top of the concatenated user-item embedding provides further flexibility and non-linearity. Formally, the MLP component of BENEFICT is defined as follows:

$$\begin{aligned} h_0 &= \left[P_u, Q_i\right]^T \\ h_1 &= ReLU(W_1 h_0 + b_1) \\ h_L &= ReLU(W_L h_{L-1} + b_L) \\ \hat{R}_{ui} &= W_{L+1} h_L + b_{L+1} \end{aligned} \tag{2}$$

where $h_0 \in \mathbb{R}^{1\times1536}$ is the concatenated user-item embedding in the shared hidden space; h_L represents the L-th MLP layer; W_L and b_L pertain to the weight matrix and bias vector of the L-th layer, respectively; and $\hat{R}_{ui}$ denotes the predicted rating that user u gives to item i. For the activation function of the MLP layers, we choose the rectified linear unit (ReLU), which generally yields better performance than other activation functions such as tanh and sigmoid (Glorot et al., 2011; He et al., 2016, 2017).

Concerning the structure, our model's MLP component follows a tower pattern where the bottom layer is the widest, and every subsequent top layer has a smaller number of neurons. The rationale behind this is that the MLP can learn more abstractive data features by utilizing fewer hidden units for the top layers (He et al., 2016). In our implementation for a three-layered MLP, the number of neurons from the bottom layer to the top layer follows this pattern: 1536 (concatenated embedding) → 768 (MLP layer 1) → 384 (MLP layer 2) → 192 (MLP layer 3) → 1 (prediction layer)

3.3 Learning

In training the model, the loss function is the mean squared error (MSE) given by this formula:

$$MSE = \frac{1}{|Tr|}\sum_{u,i\in Tr}(R_{ui} - \hat{R}_{ui})^2 \tag{3}$$

where Tr refers to the training samples or instances, and R_{ui} is the ground-truth rating given by user u to item i. Moreover, we employ the Adaptive Moment Estimation with weight decay or AdamW (Loshchilov and Hutter, 2018) to optimize the loss function. Based on the original Adam optimizer, AdamW also leverages the power of adaptive learning rates during training. This makes the selection of a proper learning rate less cumbersome that consequently leads to faster convergence (Chen et al., 2018). Unlike Adam, AdamW implements a weight decay fix, a regularization technique that prevents weights from growing too huge and is proven to yield better training loss and generalization error (Loshchilov and Hutter, 2018).

3.4 Explanation Generation

The stack of BERT's Transformer encoders also provides sets of self-attention weights that a token gives to every token found in the review text. We are particularly interested in the attention that `[CLS]` gives to each review token using the twelfth layer's multiple attention heads. Given an input sequence of tokens F_{uj} produced by WordPiece tokenization from review V_{uj}, a set of attention weights is represented as:

$$\alpha_{\texttt{[CLS]},uj} = \{\alpha_1^k(F_{uj}), \alpha_2^k(F_{uj}), ..., \alpha_g^k(F_{uj})\} \tag{4}$$

where k is the specific attention head in a particular encoder layer, and α_g^k is the attention that `[CLS]` gives to the g-th WordPiece token over the input sequence F_{uj}. There are 12 attention heads in an encoder layer which translate to 12 different attention weights that each token receives from the `[CLS]` token. For a given token g, the following formula is applied to compress the weights into a single value:

$$ComAtt_g = \sum_{k=1}^{12}\alpha_g^k(F_{uj}) \tag{5}$$

We then reformulate the task of generating explanations as a fixed-length MSP. In its vanilla sense, MSP selects a segment of consecutive array elements (i.e., a contiguous subarray of tokens) that has the maximum possible sum over all other segments (Bae, 2007). In this paper, we introduce constraint N to the MSP; N is a fixed value that pertains to the length of the explanation. Formally, the set of compressed attention weights per review is given by the following array:

$$A_{uj} = [ComAtt_1, ComAtt_2, ..., ComAtt_g] \tag{6}$$

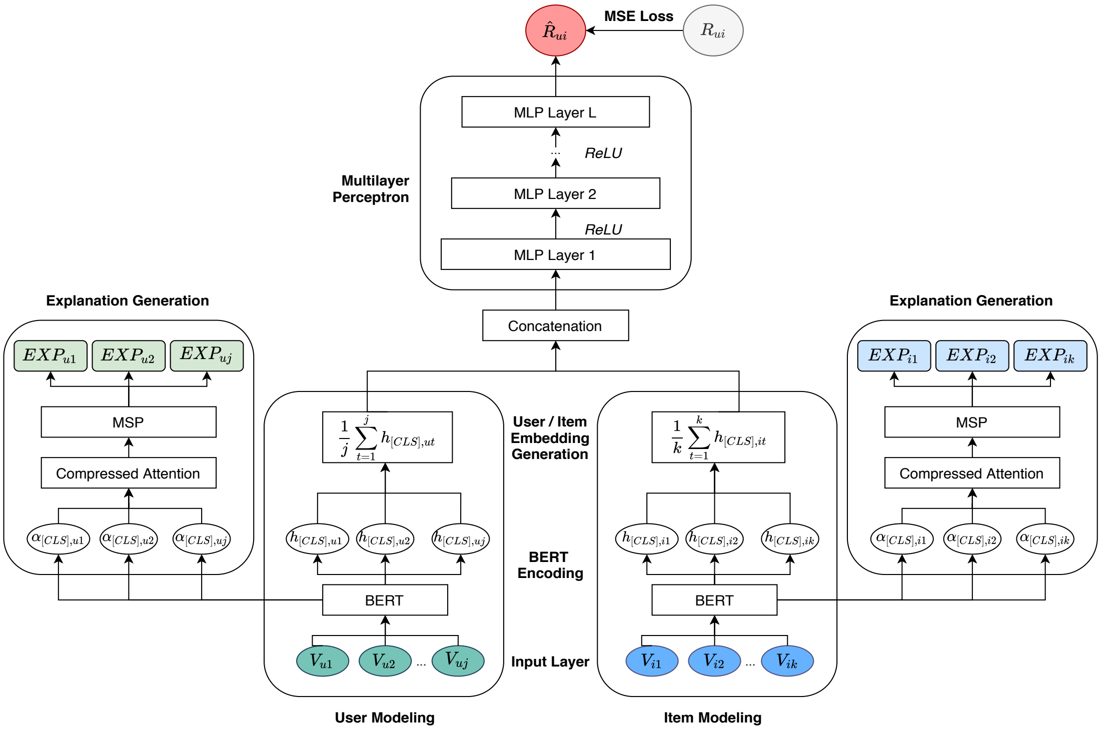

Figure 1: The proposed BENEFICT architecture.

Dataset	#Reviews	#Users	#Items
Toys and Games	167,597	19,412	11,924
Digital Music	64,706	5,541	3,568
Yelp-Dense	159,114	8,919	7,122
Yelp-Sparse	229,907	45,981	11,537

Table 1: Statistics summary of the datasets.

The goal is to find token indices x and y that maximize:

$$\sum_{t=x}^{y} A_{uj}[t] \tag{7}$$

This is subject to the requirements that $1 \leq x < y \leq 256$ and $(y - x) + 1 = N$. Finally, the generated explanation for review V_{uj} is represented as:

$$EXP_{uj} = Concat(F_{uj,x}, F_{uj,x+1}, ..., F_{uj,y}) \tag{8}$$

4 Experiments

In this section, we perform relevant experiments intending to answer the following research questions:

RQ1: Does BENEFICT outperform other state-of-the-art recommender models?

RQ2: What is the optimal configuration for learning user-item interactions?

RQ3: Can our model produce explanations acceptable to humans?

4.1 Datasets and Experimental Settings

Table 1 summarizes the four public datasets from different domains used in our study. Two of these datasets are Amazon 5-core[1]: **Toys and Games**, which consists of nearly 168 thousand reviews, and **Digital Music**, which contains about 65 thousand reviews (McAuley et al., 2015). These datasets are said to be 5-core wherein users and items have five reviews each. We also utilize **Yelp**[2], a large-scale dataset for restaurant feedback and ratings. We both employ its original, sparse version and its 5-core, dense version with about 160 thousand and 230 thousand reviews, respectively. The ratings in all datasets are in the range of [1, 5]. We randomly split each dataset of user-item pairs into training (80%), validation (10%), and test (10%) sets. In our experiments, we perform an exhaustive grid search over the following hyperparameters: number of epochs [1, 20] and number of MLP layers [0, 3]. The learning rate and weight decay are both set to

[1] http://jmcauley.ucsd.edu/data/amazon/

[2] https://github.com/danielfrg/kaggle-yelp-recruiting-competition

Model	Toys and Games	Digital Music	Yelp-Dense	Yelp-Sparse	Average
DeepCoNN	0.8971	0.8972	1.0311	1.2006	1.0065
NARRE	0.8840	0.8997	1.0312	1.1770	0.9979
BENEFICT	**0.8348**	**0.8750**	**0.9963**	**0.9764**	**0.9206**
ΔBENEFICT	5.57%	2.47%	3.38%	17.04%	7.11%

Table 2: RMSE comparison of the recommender models. The best RMSE values are highlighted in bold. The last row shows the improvement gained by BENEFICT against the better performing baseline.

0.001. Due to memory limitations, the batch size is fixed at 32. We select the model configuration (i.e., a grid point) with the best root mean square error (RMSE) on the validation set. We use the test set for evaluating the model's final performance.

4.2 Baselines and Evaluation Metric

To validate the effectiveness of BENEFICT, we select two other state-of-the-art models as baselines:

- **DeepCoNN** (Zheng et al., 2017): It is a deep collaborative neural network model based on two parallel CNNs to learn user and item feature vectors in a joint manner.
- **NARRE** (Chen et al., 2018): Similar to DeepCoNN, it is a neural attentional regression model that integrates two parallel CNNs and an attention mechanism to model latent features.

Afterward, we calculate the RMSE, a widely used metric for rating prediction, to evaluate the models' respective performances.

$$RMSE = \sqrt{\frac{1}{|Ts|} \sum_{u,i \in Ts} (R_{ui} - \hat{R}_{ui})^2} \quad (9)$$

In the formula, Ts denotes the test samples or instances of user-item pairs.

4.3 Prediction Results and Discussion

Table 2 reports the RMSE values of BENEFICT and the two baselines, with the last row (represented by ΔBENEFICT) indicating the improvement gained by our model compared with the better baseline. The results show that BENEFICT consistently outperforms the baselines across all datasets; our model has an average RMSE score of 0.9206, as opposed to 1.0065 and 0.9979 for DeepCoNN and NARRE, respectively. On average, this has resulted in the improvement gained by BENEFICT of nearly 7%. These results validate our hypothesis that using BERT-derived embeddings and representations, considered to be more semantically meaningful than their traditional counterparts, can significantly improve rating prediction accuracy and that BERT can likewise offset the limitations of mainstream word embeddings and CNN.

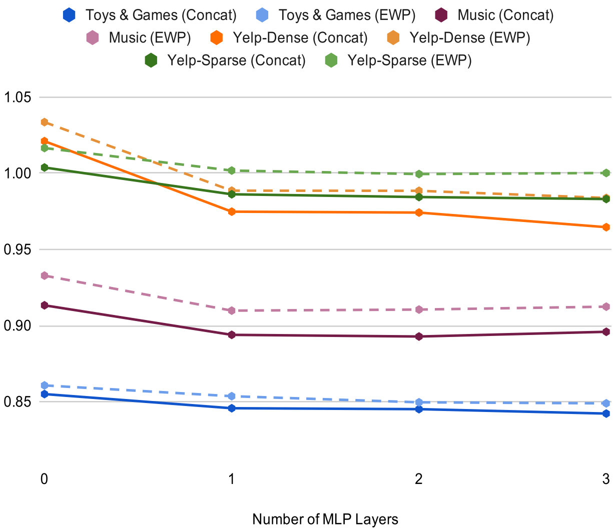

Figure 2: RMSE comparison of BENEFICT variants using different user-item interaction functions. The solid lines pertain to the concatenation-MLP interaction function. On the other hand, the broken lines refer to the interaction function based on the element-wise product (EWP) and MLP.

Moreover, the rationale of employing two versions of Yelp is to compare the recommender models' performances on both dense and sparse datasets. As illustrated in the fourth and fifth columns of Table 2, both the RMSE values of DeepCoNN and NARRE worsen when they attempt to perform predictions on the original, sparse Yelp. For DeepCoNN, from the dense version's RMSE of 1.0311, it increases to 1.2006. The same is

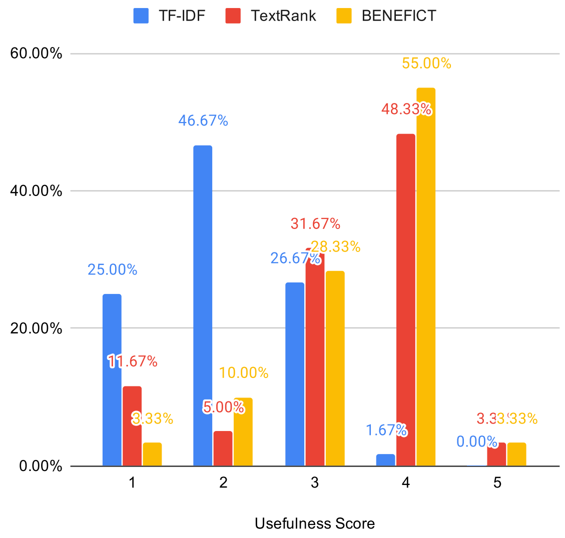

Figure 3: Distribution of the judges' given usefulness scores based on US1.

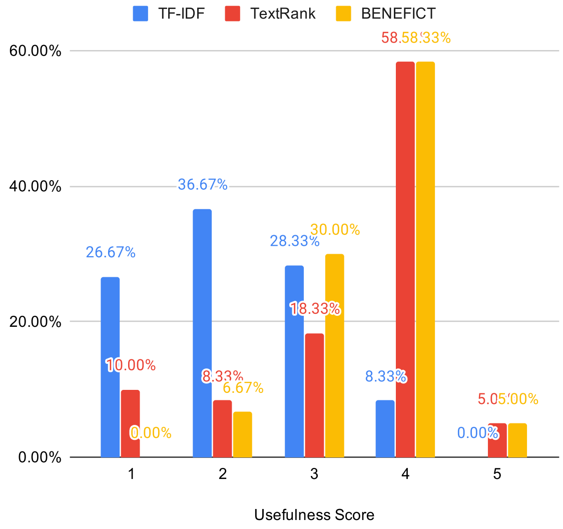

Figure 4: Distribution of the judges' given usefulness scores based on US2.

also true for NARRE, whose RMSE increases to 1.1770 from 1.0312. Interestingly, BENEFICT produces an entirely different observation; its RMSE decreases to 0.9764 from 0.9963. Our model's improvement is 17.04%, greater than ΔBENEFICT for the three other datasets. We attribute these findings to the greater amount of information in Yelp-Sparse that can be successfully utilized by BENEFICT for modeling reviews. It should be noted that Yelp-Sparse has nearly 230 thousand reviews, while Yelp-Dense has almost 160 thousand. In conclusion, these results provide evidence that our model is best equipped and capable of performing predictions regardless of a dataset's inherent sparsity or density.

4.3.1 Optimal Interaction Function

BENEFICT employs an MLP above the concatenated user-item embeddings in the shared hidden space. We compare it against another variant of our model, which utilizes an MLP on top of the element-wise product of user and item representations. We examine their performances using a different number of hidden layers [0, 3]. It should be noted that an MLP with zero layers pertains to the shared hidden space's direct projection to the prediction layer.

Figure 2 demonstrates that BENEFICT's utilization of concatenation exceeds the element-wise product by a significant margin across all MLP layers and datasets. This result verifies the positive effect of feeding the concatenated features to the MLP to learn user-item interactions. Furthermore, consistent with the findings of He et al. (2017), stacking more layers is indeed beneficial and effective for neural explicit collaborative filtering as well. There appears to be a trend: increasing the hidden layers implies decreasing (and better) RMSE values. Simply projecting the shared hidden space to the prediction layer is insufficient and weak, as evidenced by its relatively high RMSE scores. On the contrary, using three MLP layers has generally resulted in the lowest RMSE scores. The only exception is with the Digital Music dataset wherein utilizing two layers produces the best RMSE value. Furthermore, even though the element-wise product is more inferior than concatenation, the former also benefits from increasing the MLP layers. In summary, all these findings validate the necessity of incorporating the MLP as an integral part of the whole BENEFICT model.

5 Explainability Study

5.1 Human Assessment of Explanations

To validate the helpfulness of BENEFICT-produced explanations in real life, we also generate possible explanations using **TF-IDF** and **TextRank**. Applying TF-IDF determines which words are more favorable or relevant in a corpus of documents (Rajaraman and Ullman, 2011). To make the assessment fair, we only select words with the top N TF-IDF scores, where the value of N is the same as the constraint introduced in BENEFICT's

Explanation	US Scores
TF-IDF: Some of the tracks were really quite ... dare I say it, catchy. And there was even a Top 30-friendly single on the album ('Only Time will tell'). But wasn't this Carl Palmer – he of the 70s triple album and serious devotee ofclassical percussionist James Blades? And wasn't this also Steve Hose – he of another 70s triple album and several serious solo albums. And hadn't John Wetton starred on the seriously serious 'Red' in 74? How could the three come together yet produce this Adult-Oriented stadium Rock?Let's not forget Palmer's beginnings in the Crazy World of Arthur Brown and Atomic Rooster. Or Wetton'sbizarre phase with Uriah Heep. And Geoff Downes was nominally half of 'Buggles', whose minimal output was unashamed pop. The style of this, Asia's debut album wasn't a million miles from UK's eponymous LP of 1978, although it was distinctly more mainstream.I like this album, the best of all the Asia output that I've heard. I would have preferred the music to be a little more ambitious; there's a sense in which it's all been concocted to maximise the commercial return, which you couldn't say of UK. But it's a good, undemanding listen.	**US1**: 1.5 **US2**: 1.5
TextRank: Some of the tracks were really quite ... dare I say it, catchy. And there was even a Top 30-friendly single on the album ('Only Time will tell'). But wasn't this Carl Palmer – he of the 70s triple album and serious devotee of classical percussionist James Blades? And wasn't this also Steve Hose....	**US1**: 2 **US2**: 2
BENEFICT: .....The style of this, Asia's debut album wasn't a million miles from UK's eponymous LP of 1978, although it was distinctly more mainstream.I like this album, the best of all the Asia output that I've heard. I would have preferred the music to be a little more ambitious; there's a sense in which it's all been concocted to maximise the commercial return, which you couldn't say of UK.....	**US1**: 4 **US2**: 4

Table 3: Sample explanations (highlighted in yellow) generated by TF-IDF, TextRank, and BENEFICT from a specific user review. The second column includes the average judge-given US1 and US2 scores.

explanation generation module. On the other hand, TextRank is a fully unsupervised, graph-based extractive summarization algorithm (Mihalcea and Tarau, 2004). Its goal is to rank entire sentences that comprise a given review text. Also, to make the assessment consistent, we only take the top sentence with a length of less than or equal to N for each review.

We then ask two human judges to evaluate a total of 90 explanations, 30 explanations each for TF-IDF, TextRank, and BENEFICT, with $N = 20$. We instruct them to score each explanation based on the following usefulness statements (US) on a five-point Likert scale, ranging from 1 (strongly disagree) to 5 (strongly agree).

US1: The explanation captures the essence of the customer's preference (like or dislike) in the review.

US2: The explanation is helpful for you or any customer to decide to purchase that particular item in the future.

We further examine the human assessment results by determining the strength of agreement between the two judges. This is done by calculating the Quadratic Weighted Kappa (QWK) statistic. It measures inter-rater agreement and is suitable for ordinal or ranked variables. The Kappa metric lies on a scale of -1 to 1, where 1 implies perfect agreement, 0 indicates random agreement, and negative values mean that the agreement is less than chance, such as disagreement. Specifically, a coefficient of 0.01-0.20 indicates slight agreement, 0.21-0.40 implies fair agreement, 0.41-0.60 refers to moderate agreement, 0.61-0.80 pertains to substantial agreement, and 0.81-0.99 denotes nearly perfect agreement (Borromeo and Toyama, 2015).

5.2 Explainability Results and Discussion

5.2.1 Overall Assessment

Figure 3 summarizes the judges' given scores on their assessment of explanations based on US1. They find that nearly 58% of BENEFICT-derived explanations capture the essence of the customer's preference (i.e., those with usefulness scores of either four or five). It is followed by TextRank, with almost 52% of its produced explanations, and TF-IDF, with only 1.67% of its generated explanations. With respect to the inter-rater agreement on US1 in Table 5, the judges express fair agreement on BENEFICT (having a Kappa value of 0.2019). On the other hand, they slightly agree with each other on both TF-IDF and TextRank, with QWK values of 0.1924 and 0.0625, respectively. As Table 4 indicates, our model has a mean usefulness score of 3.45, better than TextRank (3.26) and TF-IDF (2.05).

Figure 4 shows the judges' assessment scores based on US2. Interestingly, the judges express that nearly 63% of the explanations generated by BENEFICT and TextRank are helpful for any future customers. Upon including the low-scoring explanations, BENEFICT is still better than TextRank; the former has a mean usefulness score of 3.61 against the latter's 3.40. Furthermore, the judges moderately agree as far as our model's generated explanation is concerned (with a Kappa value of 0.4705). At the same time, they express less than chance agreement for TextRank (obtaining a Kappa value of -0.0073). This statement means that the large majority of TextRank's high assessment scores come from one judge alone. Lastly, the judges observe that only 8.33% of the explanations from TF-IDF are helpful, with a mean usefulness score of 2.18 and a QWK value of 0.1921, which implies their slight agreement.

These results indicate that BENEFICT's explanation generation module can effectively provide useful explanations that capture the essence of the customer's preference and help future customers make purchasing decisions.

5.2.2 Specific Example Comparison

Given an example, we highlight words that serve as the explanations in Table 3. The explanation produced by TF-IDF can capture a few important words, such as *unashamed* and *undemanding*. However, due to its bag-of-words property, it includes several other unnecessary words that may not contribute to the explanation. Therefore, the judges do not find it to be helpful. Next, the TextRank-generated explanation also does not appear to capture the essence of the user's like or dislike. It does not seem useful for customers to decide whether to purchase that item in the future. Still, the judges give TextRank higher usefulness scores than TF-IDF, even though the latter captures more adjectives and important words. We attribute this to human's natural bias toward less noisy sentences that express complete thoughts. Lastly, the BENEFICT-produced explanation conveys a near-complete thought; take note that it is not a sentence but a segment of contiguous tokens that maximize the sum of attention weights. This enables BENEFICT to capture important phrases such as *like this album* and *the best of all*. Hence, the judges agree that it captures the essence of the customer's preference and helps customers make purchasing decisions in the future.

Method	US1 Mean	US2 Mean
TF-IDF	2.05	2.18
TextRank	3.26	3.40
BENEFICT	3.45	3.61

Table 4: Mean usefulness scores of explanations assessed by the judges, based on US1 and US2.

Method	US1 QWK	US2 QWK
TF-IDF	0.1924	0.1921
TextRank	0.0625	-0.0073
BENEFICT	0.2019	0.4705

Table 5: The strength of inter-judge agreement for both US1 and US2 given by the QWK values.

6 Conclusion and Future Work

We have successfully implemented a novel recommender model that uniquely integrates BERT, MLP, and MSP. BENEFICT's predictive capability is validated by experiments performed on Amazon and Yelp datasets, consistently outperforming other state-of-the-art models. Moreover, its explanation generation capability is verified by human judges. We argue that our work offers an avenue to help bridge the research gap between accuracy and explainability. In the future, we will consider incorporating other neural components, such as attention mechanisms, in improving the user-item modeling process. We also intend to enhance the expressiveness and the overall quality of the generated explanations.

Acknowledgment

First and foremost, we extend our gratitude to the hardworking anonymous reviewers for their valuable insights and suggestions. Likewise, we sincerely thank the judges in our explainability study, Dr. Ria Mae Borromeo and Ms. Verna Banasihan, for their time and participation.

References

Sung Eun Bae. 2007. Sequential and parallel algorithms for the generalized maximum subarray problem.

Ting Bai, Ji-Rong Wen, Jun Zhang, and Wayne Xin Zhao. 2017. A neural collaborative filtering model with interaction-based neighborhood. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1979–1982.

Ria Mae Borromeo and Motomichi Toyama. 2015. Automatic vs. crowdsourced sentiment analysis. In *Proceedings of the 19th International Database Engineering & Applications Symposium*, pages 90–95.

Rose Catherine and William Cohen. 2017. Transnets: Learning to transform for recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 288–296.

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

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1661–1670. ACM.

Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer product-based neural collaborative filtering. *arXiv preprint arXiv:1808.03912*.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. 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.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Andriy Mnih and Russ R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264.

Cataldo Musto, Marco de Gemmis, Giovanni Semeraro, and Pasquale Lops. 2017. A multi-criteria recommender system exploiting aspect-based sentiment analysis of users' reviews. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 321–325. ACM.

Georgina Peake and Jun Wang. 2018. Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2060–2069. ACM.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273.

Anand Rajaraman and Jeffrey David Ullman. 2011. *Mining of massive datasets*. Cambridge University Press.

Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. 2017. Social collaborative viewpoint regression with explainable recommendations. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 485–494. ACM.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? *arXiv preprint arXiv:1905.05583*.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-pointer co-attention networks for recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2309–2318.

Qianqian Wang, Si Li, and Guang Chen. 2018a. Word-driven and context-aware review modeling for recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1859–1862.

Xianchen Wang, Hongtao Liu, Peiyi Wang, Fangzhao Wu, Hongyan Xu, Wenjun Wang, and Xing Xie. 2019. Neural review rating prediction with hierarchical attentions and latent factors. In *International Conference on Database Systems for Advanced Applications*, pages 363–367. Springer.

Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2018b. Tem: Tree-enhanced embedding model for explainable recommendation. In *Proceedings of the 2018 World Wide Web Conference*, pages 1543–1552. International World Wide Web Conferences Steering Committee.

Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item silk road: Recommending items from information domains to social users. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 185–194.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Alan Zakbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728.

Yongfeng Zhang, Qingyao Ai, Xu Chen, and W Bruce Croft. 2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1449–1458.

Yongfeng Zhang and Xu Chen. 2018. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192*.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92. ACM.

Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. 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.