Leveraging Review Properties for Effective Recommendation

Xi Wang
University of Glasgow
Glasgow, UK
x.wang.6@research.gla.ac.uk

Iadh Ounis
University of Glasgow
Glasgow, UK
iadh.ounis@glasgow.ac.uk

Craig Macdonald
University of Glasgow
Glasgow, UK
craig.macdonald@glasgow.ac.uk

ABSTRACT

Many state-of-the-art recommendation systems leverage explicit item reviews posted by users by considering their usefulness in representing the users' preferences and describing the items' attributes. These posted reviews may have various associated properties, such as their length, their age since they were posted, or their rating of the item. However, it remains unclear how these different review properties contribute to the usefulness of their corresponding reviews in addressing the recommendation task. In particular, users show distinct preferences when considering different aspects of the reviews (i.e. properties) for making decisions about the items. Hence, it is important to model the relationship between the reviews' properties and the usefulness of the reviews while learning the users' preferences and the items' attributes. In this paper, we propose to model the reviews with their associated available properties. We introduce a novel review properties-based recommendation model (RPRM) that learns which review properties are more important than others in capturing the usefulness of reviews, thereby enhancing the recommendation results. Furthermore, inspired by the users' information adoption framework, we integrate two loss functions and a negative sampling strategy into our proposed RPRM model, to ensure that the properties of reviews are correlated with the users' preferences. We examine the effectiveness of RPRM using the well-known Yelp and Amazon datasets. Our results show that RPRM significantly outperforms a classical and five existing state-of-the-art baselines. Moreover, we experimentally show the advantages of using our proposed loss functions and negative sampling strategy, which further enhance the recommendation performances of RPRM.

ACM Reference Format:
Xi Wang, Iadh Ounis, and Craig Macdonald. 2021. Leveraging Review Properties for Effective Recommendation. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3442381.3450038

1 INTRODUCTION

In recent years, there has been an increase in the amount of available information and interaction choices online. As a consequence, recommender systems are increasingly being deployed in various platforms to alleviate the complexity of decision making for users and help them to find their desired items. Several studies [4, 53] focused on leveraging the item reviews posted by users. For example, Li et al. [19] modelled the users' dynamic preferences by first aggregating the reviews of users and the reviews of the items they interacted with using the reviews' temporal order. Next, they converted these reviews into embedding vectors to represent the users' preferences. However, not all reviews can be useful to represent the users' preferences and items' attributes [4]. Instead, by estimating the usefulness of reviews, the recommendation models can focus on those valuable reviews among the large volume of available information, thereby leading to an improved recommendation performance [2, 12]. Moreover, the performances of review-based recommendation models can be limited if they capture the users' preferences and items' attributes by solely using the review text [36]. Hence, many studies aimed to incorporate the usefulness of reviews in review-based recommendation [2, 4]. Other approaches have used an attention mechanism to model the usefulness of reviews [12] or those portions of the textual content of reviews that contribute most to the recommendation performances [6]. However, we argue that a limitation of such approaches is that they capture the usefulness of reviews by relying on historical data, which often do not generalise to reviews that are unseen by the trained model [42].

To address the limitation above, we propose to consider *review properties* to model the usefulness of reviews. Indeed, the reviews posted by users on items have a corresponding set of properties, such as their length, the number of days since they were posted (i.e. age) or their writing style. These review properties are associated with the historical reviews as well as the unseen reviews by the trained model. A number of studies [32, 44] have previously attempted to leverage the review properties when making recommendations. The underlying premise of such studies is that the review properties encapsulate rich information about both the users' preferences and the items' attributes.

In particular, each review property can bring useful insights about the users' preferences and the items' attributes. Therefore, by integrating such review properties into the review modelling process, a review-based recommendation model could also encapsulate the usefulness of reviews when capturing the users' preferences and items' attributes. In the literature, a number of review properties have been used as side/contextual information to enrich the user-item interactions when addressing recommendation tasks. For instance, the geographical property of reviews have been used to capture those venues visited by a user or to estimate the users' locations when making local recommendations [22, 24]. The temporal property of reviews has been frequently leveraged in sequential recommendation to predict the next actions of users [24, 47, 52]. However, these studies do not consider the review properties to examine the usefulness of reviews so as to improve the recommendation performances. In particular, it remains unclear how these

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450038

different review properties contribute to estimating the usefulness of their corresponding reviews for effective recommendation. We address this limitation by considering various review properties and examining their actual effectiveness in capturing the reviews' usefulness in addressing the recommendation task. In [39], Sussman et al. proposed the users' adoption of information framework, which showed that each user follows a particular scheme or strategy in using different properties or aspects of the review information and hence each user makes different interaction decisions. For example, according to the Elaboration Likelihood Model (ELM) [11], users can follow two strategies to process the posted reviews, namely the *central route* and the *peripheral route*. Users that follow the central route show stronger willingness in processing in-depth information (e.g. the descriptions of items in the reviews), while users who adopt the peripheral route frequently use the overall ranking or rating as key factors in making their decisions. The various reviews posted by the users differ in their characteristics (e.g. length, language, details). Such differences can be described by the reviews' properties, and are correlated to the level of the users' adoption of information [3, 30, 46]. Therefore, we argue that a model can better capture the users' preferences by learning how users use the reviews and by examining their preferences on different properties of the reviews.

In this paper, we model the importance of different review properties in capturing the usefulness of reviews and learning the users' preferences and the items' attributes. We propose a novel review property-based neural network model (RPRM) to effectively address the recommendation task. RPRM investigates the usage of review properties to model the usefulness of reviews and aims to enhance the ability of the recommendation model in capturing the usefulness of reviews and the users' adoption of information. In particular, RPRM uses six commonly encountered review properties in recommendation scenarios to encode the usefulness of reviews, including the age of the reviews, the length of the reviews, the reviews' associated ratings, the number of helpful votes associated to the reviews, the probability of the reviews being helpful and the sentiment expressed in the reviews. Note that in this paper, 'helpfulness' refers to whether users found the reviews helpful for their task, for example by capturing if they voted for them to be helpful. Different review properties can have various importance levels in capturing the usefulness of reviews for different users and items. Moreover, according to the aforementioned users' adoption of information framework [39], the properties of reviews are correlated to the level of users' adoption of information. This suggests that users tend to prefer items whose associated useful reviews capture the same important properties as those the users prefer. Therefore, we also propose two loss functions and a negative sampling strategy that aim to reward the situation where a user and the interacted items agree on the most important properties while penalising the situation where the user disagrees with the negative sampled items on the most important properties.

The main contributions[1] of this paper are as follows:

(1) We propose a novel review-based recommendation model, RPRM, which leverages the usefulness of reviews to address the recommendation task. To the best of our knowledge, this is the first work that integrates the review properties in estimating the reviews' usefulness in a recommendation model through leveraging how users make use of such reviews in their interaction with the system. **(2)** Inspired by the users' adoption of information framework, we propose two loss functions and one negative sampling strategy that model the agreement on the importance of review properties between the users and items. **(3)** We show that RPRM significantly outperforms one classical and five existing state-of-the-art recommendation approaches on the commonly-used Amazon and Yelp datasets. **(4)** We show that our proposed loss functions and the negative sampling strategy can further enhance the recommendation performances of RPRM across the two used datasets.

[1] Our source code is available at: https://github.com/wangxieric/RPRM.

2 RELATED WORK

We briefly discuss three bodies of related work, namely recommendation approaches based on reviews, recommendation approaches leveraging the use of review properties, and work investigating users' behaviour while interacting with information.

2.1 Review-based Recommendations

The main objective of applying a recommendation model is to recommend suitable items that the users might be interested in, based on observing the users' behaviour and estimating the users' preferences. User-generated reviews encapsulate rich semantic information such as the possible explanation of the users' preferences and the description of specific item attributes [5]. Therefore, many recommendation models have aimed to leverage these reviews to construct user/item representations and to address the recommendation task [1, 7, 15, 19, 27]. In addition, a number of previous review-based recommendation approaches captured the semantic similarity between the review content [4, 53], which allows to encode additional relationships among the users and items, allowing to better suggest items the users might be interested in. Indeed, the posted reviews by users are valuable in modelling the interactions among users and items from a textual semantic perspective. However, the quality and usefulness of the reviews markedly vary with the increasing amount of users and the available reviews they post online. Therefore, Chen et al. [4] applied an attention mechanism to estimate the usefulness of different reviews. Unlike previous work [2, 4], which used an attention mechanism to learn the usefulness of reviews, we argue that the review properties can be directly leveraged within an attention mechanism to effectively capture the usefulness of reviews. Moreover, there are many existing approaches [24, 32, 44] that extract the review properties and integrate them as side or contextual information to enhance the recommendation performance. However, unlike our work in this paper, such approaches do not make use of the reviews themselves. In the following, we further describe such approaches using the properties as side information.

2.2 Recommendations using Review Properties

Various existing approaches [21, 32, 33] aimed to leverage different review properties as side information to model the users' behaviour or the items' attributes. For example, Raghavan et al. [32] leveraged the extent to which a review is helpful to measure the reliability of the associated users' ratings and to incorporate such reliability scores into a recommendation model. The geographical property

of the users' reviews [22, 31] has also been well studied in venue recommendation models. Another property is whether a review expresses a sentiment. For instance, Wang et al. [44] replaced the explicit ratings provided by users with the review sentiment scores to enhance the recommendation performance. The temporal and age properties of reviews have been integrated into various recommendation models, especially into the sequential recommendation models [24, 47, 52]. For example, Manotumruksa et al. [24] encoded the temporal information in a recurrent neural network to model the users' dynamic preferences in the venue recommendation task. In particular, unlike previous works that have used review properties solely as side information in a recommendation model, in this paper we instead use them to estimate a given review's usefulness in enhancing the performance of a recommendation model. Moreover, the use of various review properties that model different aspects of the reviews has been shown to be correlated with the users' adoption of information [3, 46].

Although these approaches extracted various review properties as contextual information in order to further enrich the collaborative interactions among users and items, they have ignored the textual information of reviews. Indeed, while collaborative approaches can be effective with rich interaction information and side information [24, 50], we argue that it is still important to use both the textual information of reviews and their associated review properties. Indeed, in this study, we postulate that leveraging the review properties (e.g. length, age, sentiment) and their relationships to the users' preferences can help the recommendation model to more accurately learn the usefulness of reviews for effective recommendation. In particular, an effective recommendation model needs to also capture the users' preferences and the items' attributes along with the usage of these reviews' properties. For instance, a user might prefer to read recent reviews on a hotel to obtain a more accurate information on its current condition and services instead of reading much older reviews. This intuition also aligns with the users' adoption of information framework described in the next section.

2.3 Users' Adoption of Information

Information adoption concerns how consumers modify their behaviour by making use of the suggestions made in online reviews [11, 39]. The communication routes and the customers' involvement in a consumer opinion sharing website might persuade a customer to visit a particular destination or purchase a specific product [41]. A number of user studies have examined various properties of reviews that influence the users' adoption of information [3, 9, 11, 30, 46]. These studies observed that the properties of reviews are correlated to the level of users' adoption of information – indeed, such correlations are important motivations for our present work. For example, Filieri and Mcleay [11] used the Elaboration Likelihood Model (ELM) [29] to group the factors and properties of reviews according to two information processing routes (i.e. the central and peripheral routes). The same authors also observed that a peripheral route-based user would prefer to process information about a product that simply has a good overall ranking. On the other hand, a central route-based user would consider in-depth information to make decisions. Furthermore, Sussman et al. [39] considered the information usefulness as a mediator between the information process and the information adoption by users and showed a strong linkage between the usefulness of the information and the users' decision making. Our work is inspired by the aforementioned users' adoption of information framework. In particular, we argue that leveraging the users' preferences in relation to the reviews' properties can improve the recommendation effectiveness by providing additional insights about the users' information processing behaviour and their personalised preferences on the items' attributes as conveyed by the reviews' properties. To the best of our knowledge, this is the first work that uses and leverages the users' adoption of information to address the recommendation task.

3 METHODOLOGY

We first introduce the recommendation task and the notations used in this paper. Next, we describe our proposed RPRM model, which leverages the reviews posted by users to enhance the recommendation task. RPRM takes into account the review properties when modelling the user/item information by learning the importance of different review properties for enriching the representations of the users' preferences and the items' attributes. RPRM accounts for the importance of the review properties for both users and items by proposing two loss functions and a negative sampling strategy that model the extent to which the users and items agree on the important review properties. For example, the agreement will be high if a user considers longer reviews to be more useful and a given item's reviews usefulness is better captured by long reviews.

3.1 Task Definition

We address the recommendation task, which aims to effectively rank items for users according to their preferences. The recommendation task involves connecting two key entity types, namely: the set of users $U = \{u_1, u_2, ..., u_N\}$ with size N and the set of items $I = \{i_1, i_2, ..., i_M\}$ with size M. To address the recommendation task, we aim to accurately estimate the users' preferences on items so that we rank the items that a given user might find the most interesting in higher ranks. To do so, we investigate the use of the reviews that the users have posted on items, as well as their associated properties. Each user u or item i has an associated set of reviews, e.g. posted by that user, C_u, or posted on that item, C_i.

Furthermore, the reviews of a user or an item can be described using k review properties $\mathbf{P} = \{P_1, P_2, ..., P_k\}$. For example, to capture the preferences of user u for a given property P_1, we estimate the corresponding review property scores for each review in the review set of user u – i.e. $P_{1,u} = \{p_{1,1}, p_{1,2}, ..., p_{1,|C_u|}\}$, where $p_{1,t}$ is the property score of the t^{th} review of user u. For example, for the length (or age) property, the score will correspond to the length of the review (or its age resp.). These scores could be computed for any property, provided that the property values are mapped into scalars in the range of [0, 1] using an adequate function. For example, the geographical property ('near' vs. 'distant'), the length property ('long' vs. 'short'), or the age of reviews ('old' vs. 'recent') can all be mapped into a scalar in the interval [0, 1]. Indeed, for the length property, a longer review will have its length property score closer to 1 than other shorter reviews. Hence, the property scores depict the usefulness of reviews. In other words, the computed property scores enable the modelling of reviews from different perspectives

and examine the relationship between the review usefulness and the review properties.

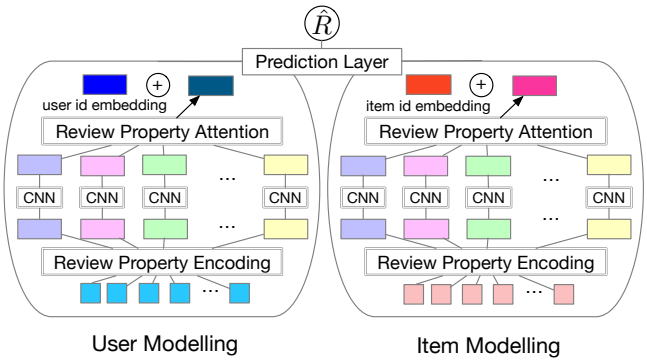


Figure 1: The Neural Network Architecture of RPRM.

3.2 The RPRM Model

To address the introduced recommendation task, we propose the novel Review Properties-based Recommendation Model (RPRM), which is a neural recommendation model that takes reviews and their associated properties into account. In particular, we use a dot-product attention mechanism to score and learn which review property is more useful in describing the usefulness of reviews and thereby are important in making good recommendations. Figure 1 presents the architecture of our proposed RPRM model. In general, RPRM is a collaborative filtering-based framework, which models the interactions between users and items. It is of note that RPRM models both the users and items using the same neural network architecture (i.e. User and Item Modelling in Figure 1). The RPRM architecture is organised into four layers, which we discuss in turn below: (1) The review property encoding layer, which combines the semantic textual representations of reviews obtained using BERT with the properties of reviews (Section 3.2.1); (2) The review embedding processing layer, which creates a low-dimensional representation of each review (Section 3.2.2); (3) The review property attention layer (Section 3.2.3), which identifies the properties of reviews that are more useful to represent the users' preferences and items' attributes; (4) We use the output from the last layer as input to the prediction layer, along with the identification embedding of a given user and item, to score the user's preferences on items (Section 3.2.4). Later, in Section 3.3, we propose and discuss new loss functions and a new negative sampling strategy to aid learning while encapsulating the properties of reviews.

3.2.1 Review Property Encoding Layer. RPRM first models the users' reviews and items' reviews. To process and summarise the semantic information of each review, we convert each review into a 768-sized embedding vector by using the pre-trained BERT model [10], which is a recent and widely used language modelling approach[2]. Next, in this layer, the model encodes the embedding vectors of the reviews with various review properties through a dot-product function. The objective of encoding the review latent vectors with different review properties is to model the usefulness of reviews from different perspectives (e.g. length, age, sentiment). Each review property can be represented by a list of normalised review property scores. These scores allow RPRM to focus on different reviews and encode the knowledge of the corresponding reviews' properties. For example, by encoding the review length property, the model can capture how reviews with different lengths can have an influence on the recommendation outcome and how the length property of reviews is associated with the user/item representations. The encoding process of a given review property can be described as follows:

$$O_{u,P_1} = [X_1 P_{1,1}, X_2 P_{1,2}, ..., X_{|C_u|} P_{1,|C_u|}] \quad (1)$$

where $X_{1,...,|C_u|}$ are the embedding vectors of the reviews of user u and $|C_u|$ is the size of their review set. In particular, Equation (1) encodes the review property P_1 for user u. After encoding k review properties, for user u, we have $O_u = [O_{u,P_1}, ..., O_{u,P_k}]$.

In this work, we use six commonly available review properties to describe the reviews from different perspectives:

• **Age:** We calculate the number of days d since a review has been posted. Then, we compute the Age score of a review to be: $p = 1 - d/max(D)$, where $max(D)$ is the age of the oldest review in the collection. In this case, a recent review is considered more useful than an older review.

•**Length:** The number of words that are included in a review.

•**Rating:** The rating associated with the review (1-5 stars).

•**Polar_Senti:** The Polar_Senti property indicates the probability of a given review being polarised (strongly positive or negative). We use a CNN classifier, which identifies reviews as being positive or negative and which has been validated as a strongly effective classifier with >95% classification accuracy in [44]. We obtain the corresponding probabilities of the positive reviews being actually positive or the negative reviews being negative[3].

•**Helpful:** The number of helpful votes given by other users to a particular review.

•**Prob_Helpful:** We classify the reviews with a state-of-the-art review helpfulness classification model [45] and obtain the probability of a given review being helpful.

In addition, for the Length, Rating and Helpful review properties, which have their property scores larger than 1, we apply the min-max normalisation to scale the property scores into [0, 1]. Note that we use the aforementioned review properties as typical review properties, which are commonly available in various datasets. However, our approach is general in that it is also possible to incorporate other review properties (e.g. the geographical and part-of-speech properties) into the proposed RPRM model.

3.2.2 Review Processing Layer. After encoding the embedding vectors of the reviews with their review property scores, we use the convolutional operators, as in other review-based deep neural network approaches [4, 53], to model the embedding vector of each review. The convolutional operators consist of m neurons, with the j^{th} neuron modelling the review embedding vector as $Z_j = ReLU(V * K_j + b_j)$, where V is the input vector, $*$ is the convolution operator with the j^{th} filter and b_j is a bias term. The ReLU activation function is applied to process the generated features. Next, each neuron j applies a sliding window over the features

[2] We use BERT for ease of integration, but any language modelling approach could be used, e.g. ALBERT [18] or RoBERTa [23].

[3] A review is positive (negative) if it has a rating ≥ 4 (≤ 2). The polarity of a 3-star review is predicted by the CNN classifier.

Z with a max pooling function to then obtain the convolutional output o_j for the corresponding neuron. Therefore, for each review, we concatenate the convolutional output from the neurons and obtain the processed embedding vector for each review (i.e. $O = [o_1, o_2, ..., o_m]$).

3.2.3 Review Property Attention Layer. In the review property encoding layer, RPRM converts the user/item modelling latent vectors into a set of latent vectors by considering various review properties. In this attention layer, the main objective is to observe which properties of reviews are more useful to represent the users' preferences and items' attributes. We hypothesise that the dot-product attention mechanism would enhance the recommendation performance of RPRM. Moreover, each user or item is associated with a review property weighted vector ϕ_u or ϕ_i with size k, where k is the number of used review properties. For a given user u, the review property attention layer is defined as follows:

$$O'_u = \frac{\sum_{t=0}^{k} \phi_{u,t} O_{u,P_t}}{k} \tag{2}$$

3.2.4 Prediction Layer. In this layer, RPRM concatenates the processed review latent vectors with the identification embedding vector of users and items to make recommendations. The final prediction of the users' preferences on items can be computed as:

$$\hat{R}_{u,i} = (O'_u \oplus V_u) \odot (O'_i \oplus V_i) \tag{3}$$

where $\oplus$ is the concatenation operation, which combines the review embedding vector O', and the identification embedding vector V. Moreover, $\odot$ denotes the element-wise product of the latent vectors between user u and item i to calculate the preference score $R_{u,i}$ of user u on item i.

3.3 Model Learning

The RPRM model addresses a ranking-based recommendation task, i.e. for a given user, it ranks first those items likely to be of interest to the user. A common and popular ranking scheme is to first apply the Bayesian Personalised Ranking (BPR) loss function [34] to optimise the model by comparing the prediction scores for users U with the positive items I^+ and the negative items I^-. The positive items are those items the user has interacted with while the negative items are sampled from those items the users did not interact with thus far. In particular, the uniform sampling strategy is commonly used to generate the negative items from the users' unseen items. We use this learning scheme as a basic setup of our proposed RPRM model.

Aside from building upon the BPR's loss function and uniform sampling for generating negative items, we propose various novel learning schemes to enhance the recommendation effectiveness. According to the users' adoption of information framework [39] that we discussed in Section 2.3, users show distinct information processing behaviour. In the recommendation scenario, users tend to have different preferences; for example some users might prefer shorter reviews while others might favour in-depth reviews that describe the advantages/disadvantages of a given item. In particular, there is a relationship between the users' behaviour and the review properties [39]. This suggests that users tend to prefer items whose associated useful reviews capture the same important properties as those the users prefer. Therefore, we propose two loss functions (i.e. $PropLoss_{uu}$ and $PropLoss_{ui}$) that reward the case of a user and the interacted items agreeing on the most important properties and penalise the case where the user disagrees with the negative sampled items on the most important properties.

In particular, based on the users' adoption of information framework, we assume that users would prefer to process information from items that have similar usefulness importance scores on the review properties. Using the information from the similarly scored items' properties, users would exhibit a higher probability of interacting with these items than with other unknown items. Therefore, both of our proposed loss functions ensure that there is an agreement on the importance of review properties between the users and their interacted items (i.e. positive items). However, $PropLoss_{uu}$ amplifies the disagreement on the importance of review properties between the users and the unseen (negative) sampled items, while $PropLoss_{ui}$ amplifies the disagreement between the interacted items and the unseen (negative) sampled items of users. These two loss functions are defined as follows:

$$PropLoss_{uu}(u, i^+, i^-) = Sim(\phi_u, \phi_{i^-}) - Sim(\phi_u, \phi_{i^+}) \tag{4}$$

$$PropLoss_{ui}(u, i^+, i^-) = Sim(\phi_{i^+}, \phi_{i^-}) - Sim(\phi_u, \phi_{i^+}) \tag{5}$$

where $Sim(.)$ is a function that measures the similarity between the weighted vectors of the review properties. Before applying the similarity function, we scale the weighting scores by dividing the scores by the sum of scores in each weighted vector to generate a discrete probability distribution of scores [0,1] over the review properties. In particular, we use the Cosine similarity (Cos) function and the Kullback–Leibler (KL) divergence measure as the (dis)similarity functions, which have shown good performances in measuring latent vector similarities [43]. Note that, since KL is a divergence measure, we use the inverse of KL to compute similarity. Furthermore, we combine the PropLoss functions with the commonly-used BPR loss function as follows:

$$\mathcal{L} = \alpha \times BPR(u, i^+, i^-) + (1 - \alpha) \times PropLoss(u, i^+, i^-) \tag{6}$$

where α controls the emphasis on the two loss functions.

We also propose a novel negative sampling strategy, called *Prop Sample*, which models the agreement on the importance of review properties between the users' interacted items and the unseen items. We argue that if the same properties are important to two items (e.g. i_1 and i_2), but a particular user interacts with item i_1 but not with item i_2, then this user shows a clearer preference for item i_1 over item i_2. Therefore, we sample negative items from each user's unseen items by selecting items that have similar review properties to those items the user has already interacted with.

For a given positive item $i^+ \in I$, we again use a similarity function $Sim()$ to calculate the similarity on the paired property weighted vectors $\phi_{i,p}$ between the positive item i^+ and all negative (unseen) items (i.e. I^-). Next, similar to the loss functions, we normalise the similarity scores across all negative items into a probability distribution. This probability distribution gives the likelihood for sampling these items as a negative instance for learning.

4 EXPERIMENTAL SETUP

In this section, we examine the performances of our proposed model and approaches on two real-world datasets. Moreover, we compare

the performance of RPRM with one classical and five existing state-of-the-art recommendation approaches. In particular, we evaluate the performances of our proposed loss functions and negative sampling strategy in addressing the following research questions:
RQ1: Does RPRM outperform the recommendation baselines on the two used datasets?
RQ2: Do the proposed loss functions, $PropLoss_{uu}$ and $PropLoss_{ui}$, improve the recommendation performances of RPRM in comparison to the classical BPR loss function?
RQ3: Does the proposed negative sampling strategy, namely *Prop Sample*, further enhance the recommendation performance of RPRM compared to the uniform sampling strategy?

4.1 Datasets & Evaluation Metrics

For answering the aforementioned research questions, we use two real-world datasets, namely the Yelp dataset[4] and the Amazon Product dataset[5] [14, 26] to examine the effectiveness of our RPRM model as well as our proposed loss functions and negative sampling strategy. The Yelp dataset includes user reviews on their top popular category (i.e. 'restaurant') and the Amazon dataset contains user reviews on products among six categories[6]. The use of various categories of the Amazon dataset allows to capture the users' preferences across different types of items/products. These two datasets have been used in several previous studies (e.g. [25, 38]). We use the Yelp dataset from the most recent round of the Yelp challenge dataset (i.e. round 13).

In our experiments, we remove cold-start users and items from both datasets, as in [8, 36], to ensure that each user and item have at least 5 associated reviews. The resulting Yelp dataset has 47k users, 16k items and 551k reviews; the Amazon dataset has 26k users, 16k items and 285k reviews. Then, following [4, 36], these two datasets are divided into 80% training, 10% validation and 10% test sets in a time-sensitive manner. In particular, we ensure that the same data split ratio applies to the interactions of each user. Next, we measure the recommendation effectiveness by examining if the items interacted with by the users in the test sets are actually chosen for recommendation by the tested models. Hence, we compute the Precision and Recall metrics at different standard rank cutoff positions (namely, P@1, P@10, and R@10) as well as Mean Average Precision (MAP), following [20, 48], to examine the effectiveness of the evaluated recommendation approaches. To test statistical significance, we apply a paired t-test, with significance level of $p < 0.05$, and use the post-hoc Tukey Honest Significant Difference (HSD) [37] at $p < 0.05$ to account for the multiple comparisons with the t-tests. In the following, we describe the experimental setup of both RPRM and the used baselines.

4.2 Model Training

We implement our proposed RPRM model and the NN-based baseline approaches (namely DREAM, CASER, DeepCoNN, JRL and NARRE) using the PyTorch framework [28]. For the setup of RPRM, in the review processing layer, as introduced in Section 3.2, we use the pre-trained BERT model [10] to convert each review into a 768-sized latent vector. However, since BERT is limited to encoding a maximum of 512 tokens, we limit the maximum review length to 512 tokens. Next, in the review property encoding layer, we use the Negative Confidence-aware Weakly Supervised (i.e. NCWS) review helpfulness classifier [45] to generate the 'Polar_Helpful' property scores, which estimates the probability of the reviews being helpful. In particular, we follow [45] in training the NCWS model using reviews from the 'food' and 'nightlife' categories of the Yelp Challenge dataset round 12[7] and on the Kindle reviews from Amazon[8]. We use NCWS to predict the 'Polar_Helpful' property scores of reviews in both the Yelp and Amazon datasets. Similarly, to generate the 'Polar_Senti' review property scores, we use a CNN-based binary sentiment classifier [16], which has been shown to have a strong classification accuracy (>95%) [44]. We then train it on 50,000 positive and 50,000 negative sentiment reviews that are sampled from the Yelp Challenge dataset round 12 to conduct sentiment classification [44]. We label the polarity of each review according to the user's posted rating, which we label as positive if the rating ≥ 4, and negative if the rating ≤ 2. This CNN classifier provides each review with its probability of carrying a strong polarised sentiment. Finally, when training our proposed RPRM model, we apply early-stopping and use the Adam optimiser [17] with a $5e^{-4}$ and $1e^{-3}$ learning rates for the Yelp and Amazon datasets, respectively. These learning rates are selected after tuning the model on the validation set, varying the learning rates between $1e^{-5}$ and $1e^{-3}$.

4.3 Baseline Approaches

We use six baselines: one classical baseline and five existing state-of-the-art recommendation approaches: **(1) BPR-MF** [34] is a traditional and commonly used recommendation baseline that uses a pairwise ranking loss function (i.e. BPR) to learn the matrix factorised interactions between users and items. **(2) DREAM** [49] encodes the age property of the reviews and models the dynamic representations of the users' preferences with a recurrent neural network. **(3) CASER** [40] is a recent approach that sequentially models the implicit user historical interactions with convolutional neural networks. It is a state-of-the-art recommendation model [13] that encodes the age property of reviews. **(4) DeepCoNN** [53] is a review-based recommendation model that jointly models users and items through a convolutional neural network. **(5) JRL** [51] is a heterogeneous recommendation model that encodes various types of information resources including product images, review text and user ratings. In particular, we implement the JRL model by only using the review text. **(6) NARRE** [4] models users and items with two parallel neural networks, both of which include a convolutional layer and an attention layer to capture the usefulness of reviews.

To ensure a fair comparison, we also apply early-stopping on all baseline approaches. Moreover, since we use a pre-trained BERT model to convert the reviews into embedding vectors for our proposed RPRM model, we also extend the DeepCoNN and NARRE baselines by using the BERT-encoded review embedding vectors. In particular, for DeepCoNN, we concatenate all reviews given by/to a single user/item and form a user/item review document. Similar to RPRM, for both approaches, we limit the maximum length of

[4] https://www.yelp.com/dataset [5] http://jmcauley.ucsd.edu/data/amazon/ [6] 'amazon instant video', 'automotive', 'grocery and gourmet food', 'musical instruments', 'office products' and 'patio lawn and garden'

[7] We use different Yelp dataset rounds, different categories & removed reviews that belong to 'restaurant' from 'food' and 'nightlife', to avoid overlaps between the NCWS and RPRM evaluation settings. [8] Again, we use a different Amazon review category for training NCWS to avoid any overlap with the RPRM evaluation.

Table 1: Recommendation performances. Significant differences w.r.t. 'No-Prop' are indicated by '*' (according to both the paired t-test and the Tukey HSD test, $p < 0.05$). 1/2/3 denote a significant difference according to both tests w.r.t. to the indicated approach. ↑ indicates that the corresponding approach is significantly outperformed by RPRM on all ranking metrics according to both tests.

Dataset	Amazon				Yelp			
Model	**P@1**	**P@10**	**R@10**	**MAP**	**P@1**	**P@10**	**R@10**	**MAP**
↑BPR-MF	0.0053*	0.0034*	0.0301*	0.0111*	0.0101*	0.0058*	0.0391*	0.0145*
↑DREAM	0.0030*	0.0008*	0.0062*	0.0029 *	0.0083*	0.0065*	0.0469*	0.0155*
↑CASER	0.0093*	0.0060*	0.0499*	0.0239 *	0.0111*	0.0083*	0.0571*	0.0229*
↑1 DeepCoNN	0.0053*2,3	0.0037*2,3	0.0343*2,3	0.0119*2,3	0.0054*2,3	0.0025*3	0.0173*2,3	0.0072*2,3
↑2 JRL	0.0041*1,3	0.0031*1,3	0.0310*1,3	0.0092*1,3	0.0043*1,3	0.0021*3	0.0135*1,3	0.0061*1,3
↑3 NARRE	0.0175*1,2	0.0066*1,2	0.0588*1,2	0.0279*1,2	0.0137*1,2	0.0087*1,2	0.0605*1,2	0.0228*1,2
↑No-Prop	0.0208	0.0088	0.0805	0.0357	0.0153	0.0099	0.0745	0.0260
Age	0.0215*	0.0089	0.0820*	0.0372*	0.0157	0.0105*	0.0756*	0.0267*
Length	0.0214	0.0089	0.0815*	0.0364*	0.0159*	0.0101	0.0726*	0.0262
Helpful	0.0218*	0.0089	0.0817*	0.0365*	0.0151	0.0100	0.0719*	0.0255
Prob-Helpful	0.0214	0.0093	0.0852*	0.0376*	0.0152	0.0103	0.0750	0.0264
Rating	0.0206	0.0087	0.0795*	0.0352	0.0160*	0.0102	0.0730*	0.0264
Polar-Senti	0.0211	0.0086	0.0783*	0.0355	0.0155	0.0102	0.0738	0.0262
RPRM	**0.0223***	**0.0095***	**0.0865***	**0.0378***	**0.0161***	**0.0104**	**0.0761***	**0.0271***

the user/item document in the DeepCoNN model, and the maximum tokens of each review in NARRE, to 512 tokens. We then fine tune every baseline model with learning rates in $[1e^{-3}, 1e^{-4}, 1e^{-5}]$ and compare our approaches with the settings that exhibited the best performances on the validation set. We evaluate the various components of our proposed RPRM model incrementally. First, we capture the effectiveness of using the review properties in a review-based recommendation model. Next, we remove the review property encoding layer in RPRM, denoted as 'No-Prop', to examine its effectiveness on our datasets. Next, we also examine the effectiveness of using each single review property from the included properties in Section 3.2. Therefore, we apply each single review property in the review property attention layer of RPRM to evaluate their effectiveness in identifying the usefulness of reviews. We denote the resulting recommendation models with the name of the corresponding review properties (e.g. 'Age' for using the Age property). Next, we examine the effectiveness of our proposed RPRM's learning schemes, namely the two loss functions ($PropLoss_{uu}$ and $PropLoss_{ui}$) and the negative sampling strategy $PropSample$, by comparing their performances with those of the commonly used BPR loss function and with uniform sampling, respectively.

5 RESULTS AND ANALYSIS

We present and analyse the results of our experiments to answer the research questions in Section 4. Our experiments focus on investigating the performance of RPRM as well as the effectiveness of our proposed loss functions and negative sampling strategies in comparison to the six strong baselines from the literature.

5.1 RQ1: Review Property-based Model Evaluation

To answer RQ1, we first examine the performances of the six baselines and compare them to the performances of our proposed RPRM model and its variants. In particular, we integrate each review property separately, before combining all of them together in the full RPRM model. The results on the two used datasets are presented in Table 1. First, we compare the performance of the RPRM model without using the review property encoding layer (namely No-Prop) to the baseline approaches from Table 1. We observe that No-Prop significantly outperforms all baseline approaches, including the state-of-the-art recommendation approaches (namely CASER, DeepCoNN and NARRE), according to both the paired t-test and the Tukey HSD test regardless of whether they use any review information. In particular, we focus on DeepCoNN, JRL and NARRE that make use of review information. Both DeepCoNN and JRL exhibit weak recommendation performances on the two used datasets with low precision, recall and MAP scores, which are lower than the traditional BPR-MF approach. The BPR-MF approach is a strong baseline and was shown recently to outperform various state-of-the-art recommendation approaches from the literature [35]. Among these three baselines, NARRE significantly outperforms both the DeepCoNN and JRL approaches according to both the paired t-test and the Tukey HSD test on the two datasets with higher evaluation scores. However, NARRE is significantly outperformed by our No-Prop variant (according to both the paired t-test and the Tukey HSD test), despite No-Prop having a simpler structure than NARRE. The effectiveness of this simple review-based recommendation approach is consistent with the conclusions in [36]. Moreover, by comparing No-Prop and DeepCoNN, we note that the only architecture difference between these two models is that No-Prop integrates the identification embedding vectors of users and items. The observed significantly enhanced performances of No-Prop over DeepCoNN on all used metrics (paired t-test and Tukey HSD test) demonstrate the benefits of using such embedding vectors to model the users' preferences and items' attributes. In summary, we find that No-Prop significantly outperforms all baseline approaches. In particular, the use of the embedding vectors, which model the users' preferences and items' attributes, explains the superior performances of both the No-Prop and NARRE models in comparison to other baseline approaches.

Next, we evaluate the effectiveness of integrating different review properties to the basic No-Prop approach to model the usefulness of reviews. The results from Table 1 show that in general, the

Table 2: Impact of the model's learning schemes on RPRM. Statistically significant differences with respect to 'RPRM-basic' are indicated by '*' (according to both the paired t-test and the Tukey HSD test, $p < 0.05$).

Dataset	Amazon				Yelp			
Model	**P@1**	**P@10**	**R@10**	**MAP**	**P@1**	**P@10**	**R@10**	**MAP**
RPRM-basic	0.0223	0.0095	0.0865	0.0378	0.0161	0.0104	0.0761	0.0271
$PropLoss_{uu}$-KL	0.0217	0.0094	0.0867	0.0381	0.0163	0.0106	0.0772*	0.0281*
$PropLoss_{uu}$-Cos	0.0211*	0.0093	0.0853*	0.0369*	0.0154*	0.0105	0.0764	0.0271
$PropLoss_{ui}$-KL	**0.0226**	**0.0097**	**0.0894***	**0.0385†**	**0.0175***	**0.0107**	**0.0788***	**0.0288†**
$PropLoss_{ui}$-Cos	0.0211*	0.0093	0.0859*	0.0382	0.0165	0.0105	0.0772*	0.0278*
PropSample-KL	0.0225	0.0093	0.0863	0.0385*	0.0163	0.0102	0.0738*	0.0268
PropSample-Cos	0.0220	0.0093	0.0855	0.0376	0.0166	0.0105	0.0767*	0.0276
$PropLoss_{ui}$-KL +*PropSample*-KL	0.0210*	0.0095	0.0871*	0.0373	0.0162	0.0100	0.0740*	0.0266
$PropLoss_{ui}$-KL +*PropSample*-Cos	0.0211*	0.0094	0.0863	0.0372*	0.0159	0.0105	0.0770*	0.0273

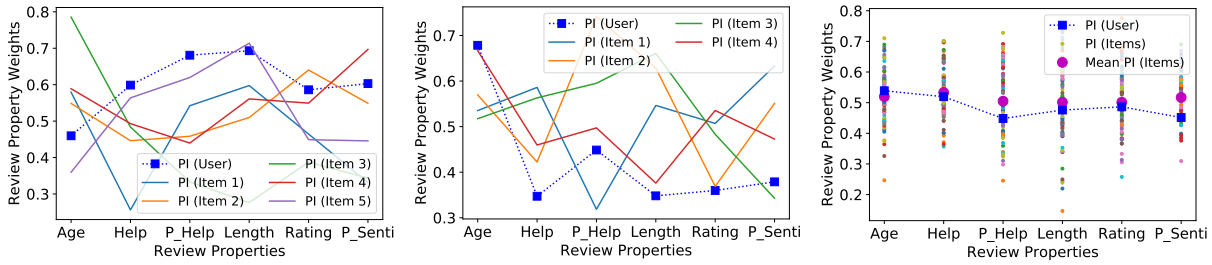
review properties can significantly improve the performances of No-Prop on both used datasets according to both the paired t-test and the Tukey HSD test. In particular, we observe that the 'Age' and 'Prob_Helpful' review properties are the two most effective properties among the six review properties we tested in capturing the usefulness of reviews and improving the recommendation effectiveness of No-Prop. The other review property-based approaches show different performances on the two datasets. For example, the 'Helpful' property enhances the performances of No-Prop on the Amazon dataset but decreases its performances on the Yelp dataset. Moreover, the 'Rating' property improves the recommendation performances of No-Prop on the Yelp dataset but not on the Amazon dataset. Therefore, these results suggest that it is more effective to selectively apply the right review properties in the recommendation model to assess the usefulness of the reviews and to leverage them in the made recommendations, which is one of the main underlying ideas of our proposed RPRM model. In particular, these results indicate the necessity of understanding the importance of different review properties on different datasets or recommendation applications. Therefore, next, we evaluate the performance of our proposed RPRM model, which integrates all six review properties and appropriately scores (or weights) the importance of different reviews' properties. The observed results for RPRM from Table 1 show that the RPRM model provides the best recommendation effectiveness on the two used datasets. Moreover, the observed performances significantly outperform both No-Prop and all the baseline approaches, including the existing state-of-the-art recommendation models (namely, NARRE, CASER and DeepCoNN), according to both the paired t-test and the Tukey HSD test. These results demonstrate the benefits of using all review properties and weighting their importance for capturing the useful reviews and their leverage in recommendation.

Therefore, in answering RQ1, we conclude that different review properties show distinct effectiveness levels in enhancing the performance of a review-based recommendation model. Among the six used review properties, the 'Age' and 'Prob_Helpful' properties are the most effective, and consistently enhance the effectiveness of the No-Prop recommendation model. Furthermore, by integrating all six review properties and weighting their importance in the full RPRM model, we observe that RPRM achieves the best performance among all evaluated approaches on both used datasets. Our results also validate our hypothesis in Section 3.2.3, namely that weighting the importance of review properties with a dot-product attention mechanism can enhance the recommendation performances.

5.2 RQ2: Effectiveness of the Proposed Loss Functions

To answer RQ2, we examine the impact of using our proposed loss functions (namely $PropLoss_{uu}$ and $PropLoss_{ui}$) using two different (dis)similarity approaches (namely the KL divergence and Cosine similarity). We also compare the RPRM model that uses our proposed loss functions with the same model using a standard ranking scheme, namely the BPR loss function and a uniform sampling strategy for generating negative items (i.e. RPRM-basic). By exploring different combinations, we have four possible model learning setups, i.e. $PropLoss_{uu}$ with KL or Cosine and $PropLoss_{ui}$ with KL or Cosine. Table 2 presents the obtained experimental results on the two Amazon and Yelp datasets for these four model learning setups. First, from Table 2, we observe that our proposed two loss functions can consistently improve the performance of the basic RPRM with the exception of the $PropLoss_{uu}$-Cos model setup on the Amazon dataset. In particular, by comparing the evaluation performances of the PropLoss-based approaches with that of the basic RPRM, we observe that $PropLoss_{ui}$-KL improves upon the recommendation performances of RPRM-basic with significantly higher MAP scores according to both the paired t-test and the Tukey HSD test on the two used datasets: 0.0378 → 0.0385 on the Amazon dataset and 0.0271 → 0.0288 on the Yelp dataset, which is significant according to both the paired t-test and the Tukey HSD test at $p < 0.05$.

Next, we compare the impact of integrating the two proposed loss functions in turn into RPRM. From the results in Table 2, we observe that the $PropLoss_{ui}$-based approaches outperform the $PropLoss_{uu}$-based approaches on both datasets. This observation suggests that, in terms of setting the importance of the used review properties, it is more effective to amplify the disagreement between the users' interacted items and the negatively sampled items than that between users and the negatively sampled items. Our results also demonstrate that leveraging the users' adoption of information framework is a promising approach. Finally, by examining the effectiveness of the two similarity measurement approaches, we observe that



(a) User A (few interactions). (b) User B (few interactions). (c) User C (many interactions).

Figure 2: The properties' importance scores of reviews for randomly selected users and their interacted items. 'Help', 'P_Help', 'P_Senti' are the abbreviations of 'Helpful', 'Prob_Helpful' & 'Polar_Senti', resp. 'PI' refers to the Properties' Importance scores.

both the KL and Cosine (dis)similarity-based approaches can outperform the RPRM-basic model when applied with the $PropLoss_{ui}$ approaches. However, KL is consistently more effective on both datasets in comparison to the Cosine similarity method and significantly better than RPRM-basic according to both the paired t-test and the Tukey HSD test. This result overall demonstrates the effectiveness of modelling the divergence between the weighting vectors of the review properties on our used datasets.

After analysing the results from Table 2, we now answer RQ2: our proposed loss functions $PropLoss_{uu}$ and $PropLoss_{ui}$ can both improve the performances of RPRM-basic. Moreover, $PropLoss_{ui}$ shows a higher effectiveness than $PropLoss_{uu}$ in enhancing the recommendation performance. We conclude that the divergence between the weighted vectors of the review properties using the KL divergence measure, $PropLoss_{ui}$, can enhance the RPRM-basic model and gives the best overall recommendation performances.

5.3 RQ3: Effectiveness of the Proposed Negative Sampling Strategy

We now examine the effectiveness of our proposed negative sampling strategy (namely *PropSample*), so as to answer RQ3. From Table 2, we observe that *PropSample* does not consistently outperform the RPRM-basic model when using the same similarity approach. In particular, the *PropSample* model with the KL divergence can improve the recommendation performance of RPRM-basic on the Amazon dataset but not on the Yelp dataset. On the other hand, the *PropSample* model that uses the Cosine similarity can improve the recommendation performance of RPRM-basic on the Yelp dataset but not on the Amazon dataset. Next, we investigate combining the *PropSample* negative sampling strategy with the best performing loss function, namely $PropLoss_{ui}$-KL (see the last two rows of Table 2). We observe that the combination of both *PropSample* and $PropLoss_{ui}$-KL with RPRM-basic does not lead to a better performance than when solely using $PropLoss_{ui}$-KL. These results might be caused by the fact that both $PropLoss_{ui}$ and *PropSample* similarly capture the importance of the review properties between the users' interacted items and the unseen items. Furthermore, *PropSample* only considers the agreement between the users' interacted (positive) and the unseen (negative) items on the important reviews' properties, which might not be sufficient to sample useful negative items. We leave the modelling of further additional information in the *PropSample* approach (e.g. the agreement on the important reviews' properties between the users and their interacted (positive) and/or unseen (negative) items) to future work.

Therefore, to answer RQ3, we conclude that *PropSample* can enhance RPRM if an adequate similarity measure is applied on each used dataset. In addition, by comparing the performances of the *Prop -Sample* and *PropLoss* approaches, our results show that the *PropLoss* loss function has more impact on the recommendation effectiveness than the negative sampling strategy, suggesting that it is more important to capture the reviews' properties importance between the users and their interacted or unseen items.

6 USERS' PROPERTY PREFERENCES

The users' adoption of information is one of the main arguments underlying our proposed RPRM model. In Section 5, we showed the effectiveness of modelling the agreement between the users and items in terms of the reviews' properties. Therefore, in this section, we use three randomly selected users to illustrate the users' preferences on different review properties and the agreement on the importance of review properties between the users and their interacted items.

To this end, Figure 2(a)-(c) plots the learned RPRM property importance scores for the review properties of three randomly selected users, say A, B & C, as well as their interacted items. The users' property importance preferences are shown using a blue dashed line with square markers; their interacted items in the test set are also shown (solid lines in Figures 2(a) and 2(b) and dots in different colours in Figure 2(c)) from the Yelp dataset. In particular, we selected users A & B as example users that have few interactions with items, to illustrate the importance scores on the review properties between the target users and their interacted items. Indeed, we select users with few interactions so as to be able to visually plot all these items in a figure. From Figure 2(a), we observe that user A shows stronger preferences for the 'Length' property (i.e. A prefers longer reviews) and that the 'Length' property is also highly weighted when determining the usefulness of reviews associated to that user's interacted items. On the other hand, for user B (Figure 2(b)), the 'Age' property is an important review property (i.e. B prefers recent reviews) to capture the review usefulness, which is similar to the high weights on 'Age' for the interacted items.

Next, since users A and B have few item interactions, they might not accurately reflect the behaviour of the general user population

in using different reviews. Therefore, we plot the learned importance scores of the review properties for a third user, C, who has interacted with a higher number of items. We also plot the mean importance scores of the review properties of his/her interacted items in Figure 2(c). From Figure 2(c), we observe that the importance scores on the review properties of user C is close to the mean importance scores on the review properties of his/her interacted items, especially on the two most important review properties ('Age' and 'Helpful'). This tells us that the 'Age' and 'Helpful' properties are important properties to observe the usefulness of reviews for both user C and his/her interacted items.

The above figures provide further evidence that users and their interacted items agree on the important review properties. We envisage that an online platform could leverage these weighting scores to customise the review presentation to different users according to their preferences for different review properties. For example, it is better to present recent reviews to user B than to user A, while user A would prefer to see longer reviews so as to obtain more information about the items' features. Our proposed RPRM model can learn the importance of the review properties to identify useful reviews and enables making review presentation decisions.

7 CONCLUSIONS

We proposed the review-based RPRM model, which leverages the importance of different review properties in capturing the usefulness of reviews thereby enhancing the recommendation performance. Inspired by the users' adoption of information framework [39], we proposed two new loss functions and a negative sampling strategy that account for the usefulness of the review properties. RPRM consistently outperformed six strong existing recommendation models across the two used Yelp and Amazon datasets. Moreover, we have shown that both of our proposed loss functions and negative sampling strategy can further improve the recommendation performances of RPRM. These results demonstrated the advantages of leveraging the agreement on the review properties' importance between users and items. Through a qualitative analysis, we have also illustrated the recommendation added-value of RPRM by examining the usefulness of several review properties for a sample of users and their interacted items. This analysis has exemplified the promise of RPRM in assisting online review platforms in customising the presentation of reviews and deploying more effective recommendation systems.

REFERENCES

[1] Amjad Almahairi, Kyle Kastner, Kyunghyun Cho, and Aaron Courville. 2015. Learning distributed representations from reviews for collaborative filtering. In *Proc. of RecSys*.

[2] Konstantin Bauman, Bing Liu, and Alexander Tuzhilin. 2017. Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. In *Proc. of SIGKDD*.

[3] Yolanda YY Chan and Eric WT Ngai. 2011. Conceptualising electronic word of mouth activity. *Marketing Intelligence & Planning* (2011).

[4] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proc. of WWW*.

[5] Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction* 25, 2 (2015), 99–154.

[6] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2019. Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In *Proc. of SIGIR*.

[7] Yifan Chen, Yang Wang, Xiang Zhao, Hongzhi Yin, Ilya Markov, and MAARTEN De Rijke. 2020. Local Variational Feature-based Similarity Models for Recommending Top-N New Items. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–33.

[8] Zhiyong Cheng, Ying Ding, Xiangnan He, Lei Zhu, Xuemeng Song, and Mohan S Kankanhalli. 2018. A^3NCF: An Adaptive Aspect Attention Model for Rating Prediction. In *Proc. of IJCAI*.

[9] Cindy Man-Yee Cheung, Choon-Ling Sia, and Kevin KY Kuan. 2012. Is this review believable? A study of factors affecting the credibility of online consumer reviews from an ELM perspective. *Journal of the Association for Information Systems* 13, 8 (2012).

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL-HLT*.

[11] Raffaele Filieri and Fraser McLeay. 2014. E-WOM and accommodation: An analysis of the factors that influence travelers' adoption of information from online reviews. *Journal of Travel Research* 53, 1 (2014), 44–57.

[12] Xinyu Guan, Zhiyong Cheng, Xiangnan He, Yongfeng Zhang, Zhibo Zhu, Qinke Peng, and Tat-Seng Chua. 2019. Attentive Aspect Modeling for Review-aware Recommendation. *ACM Transactions on Information Systems (TOIS)* 37, 3 (2019), 28.

[13] Qing Guo, Zhu Sun, Jie Zhang, and Yin-Leng Theng. 2020. An Attentional Recurrent Neural Network for Personalized Next Location Recommendation. In *Proc. of AAAI*.

[14] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proc. of WWW*.

[15] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proc. of CIKM*.

[16] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proc. of EMNLP*.

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proc. of ICLR*.

[18] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *Proc. of ICLR*.

[19] Chenliang Li, Xichuan Niu, Xiangyang Luo, Zhenzhong Chen, and Cong Quan. 2019. A Review-Driven Neural Model for Sequential Recommendation. In *Proc. of IJCAI*.

[20] Yuqi Li, Weizheng Chen, and Hongfei Yan. 2017. Learning Graph-based Embedding For Time-Aware Product Recommendation. In *Proc. of CIKM*.

[21] Guang Ling, Michael R Lyu, and Irwin King. 2014. Ratings meet reviews, a combined approach to recommend. In *Proc. of RecSys*.

[22] Wei Liu, Zhi-Jie Wang, Bin Yao, and Jian Yin. 2019. Geo-ALM: POI Recommendation by Fusing Geographical Information and Adversarial Learning Mechanism.. In *Proc. of IJCAI*.

[23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019).

[24] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2018. A contextual attention recurrent architecture for context-aware venue recommendation. In *Proc. of SIGIR*.

[25] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proc. of RecSys*.

[26] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proc. of SIGKDD*. 785–794.

[27] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proc. of EMNLP-IJCNLP*. 188–197.

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proc. of NeurIPS*.

[29] Richard E Petty and John T Cacioppo. 2012. *Communication and persuasion: Central and peripheral routes to attitude change*. Springer Science & Business Media.

[30] Hamed Qahri-Saremi and Ali Reza Montazemi. 2019. Factors Affecting the Adoption of an Electronic Word of Mouth Message: A Meta-Analysis. *Journal of Management Information Systems* 36, 3 (2019).

[31] Tieyun Qian, Bei Liu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2019. Spatiotemporal representation learning for translation-based POI recommendation. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–24.

[32] Sindhu Raghavan, Suriya Gunasekar, and Joydeep Ghosh. 2012. Review quality aware collaborative filtering. In *Proc. of RecSys*.

[33] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. 2017. Social collaborative viewpoint regression with explainable recommendations. In *Proc. of WSDM*.

[34] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proc. of UAI*.
[35] Steffen Rendle, Walid Krichene, Li Zhang, and John R. Anderson. 2020. Neural Collaborative Filtering vs. Matrix Factorization Revisited. In *Proc. of RecSys*.
[36] Noveen Sachdeva and Julian McAuley. 2020. How Useful are Reviews for Recommendation? A Critical Review and Potential Improvements. In *Proc. of SIGIR*.
[37] Tetsuya Sakai. 2018. *Laboratory Experiments in Information Retrieval - Sample Sizes, Effect Sizes, and Statistical Power*. Springer.
[38] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proc. of RecSys*.
[39] Stephanie Watts Sussman and Wendy Schneier Siegal. 2003. Informational influence in organizations: An integrated approach to knowledge adoption. *Information systems research* 14, 1 (2003), 47–65.
[40] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proc. of WSDM*.
[41] Liang Rebecca Tang, Soocheong Shawn Jang, and Alastair Morrison. 2012. Dual-route communication of destination websites. *Tourism Management* 33, 1 (2012), 38–49.
[42] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proc. of ICLR*.
[43] Xi Wang, Anjie Fang, Iadh Ounis, and Craig Macdonald. 2019. Evaluating Similarity Metrics for Latent Twitter Topics. In *Proc. of ECIR*.
[44] Xi Wang, Iadh Ounis, and Craig Macdonald. 2019. Comparison of Sentiment Analysis and User Ratings in Venue Recommendation. In *Proc. of ECIR*.
[45] Xi Wang, Iadh Ounis, and Craig Macdonald. 2020. Negative Confidence-Aware Weakly Supervised Binary Classification for Effective Review Helpfulness Classification. In *Proc. of CIKM*.
[46] Chenghuan Wu and David R Shaffer. 1987. Susceptibility to persuasive appeals as a function of source credibility and prior experience with the attitude object. *Journal of personality and social psychology* 52, 4 (1987), 677.
[47] Jibang Wu, Renqin Cai, and Hongning Wang. 2020. Déjà vu: A Contextualized Temporal Attention Mechanism for Sequential Recommendation. In *Proc. of WWW*.
[48] Jheng-Hong Yang, Chih-Ming Chen, Chuan-Ju Wang, and Ming-Feng Tsai. 2018. HOP-rec: high-order proximity for implicit recommendation. In *Proc. of RecSys*.
[49] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proc. of SIGIR*.
[50] Jia-Dong Zhang and Chi-Yin Chow. 2015. GeoSoCa: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *Proc. of SIGIR*.
[51] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W Bruce Croft. 2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In *Proc. of CIKM*.
[52] Pengpeng Zhao, Haifeng Zhu, Yanchi Liu, Jiajie Xu, Zhixu Li, Fuzhen Zhuang, Victor S Sheng, and Xiaofang Zhou. 2019. Where to go next: A spatio-temporal gated network for next poi recommendation. In *Proc. of AAAI*.
[53] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proc. of WSDM*.