Recommending novel and relevant reviews to expand users' knowledge about a product

Edgar Ceh-Varela
Department of Computer Science
New Mexico State University
Las Cruces, NM, USA
eceh@nmsu.edu

Huiping Cao
Department of Computer Science
New Mexico State University
Las Cruces, NM, USA
hcao@cs.nmsu.edu

***Abstract*—Most e-commerce websites (e.g., Amazon and TripAdvisor) show their users an initial set of useful product reviews. These reviews allow users to form a general idea about the product's characteristics. The usefulness of a review is mainly based on a score that the website users provide. Studies have shown that this score is not a good indicator of a review's actual helpfulness. Nonetheless, most past works still use it to classify a review as helpful or not. With the growing number of reviews, finding those helpful ones is a challenging task. In this work, we propose *NovRev*, a new unsupervised approach to recommend a personalized subset of unread useful reviews for those users looking to increase their knowledge about a product. *NovRev* considers an initial set of reviews as a context and recommends reviews that increase the product's information. We have extensively tested *NovRev* against five baseline methods, using eight real-life datasets from different product domains. The results show that *NovRev* can recommend novel, relevant, and diverse reviews while covering more information about the product.**

***Index Terms*—reviews, recommender system, knowledge, aspects**

I. Introduction

In e-commerce, reviews are helpful to users because they contain essential product knowledge, valuable customer preferences, and insights for using the products. Most e-commerce websites (e.g., Amazon and TripAdvisor) let their users write reviews about products or services offered on them. These reviews allow users to evaluate other users' experience with a product and rate this experience with a helpfulness vote. The importance of these reviews is relevant, 79% of Amazon[1] consumers read reviews before making a purchase, and 83% of TripAdvisor[2] users indicated that reviews help them pick the right hotel. However, a large amount of these reviews is overwhelming to users.

A review's helpfulness score is calculated based on the votes that users give to the review. However, studies [1] have found that this score is not useful. This score is mainly biased towards reviews that are more positive, more extensive or have a higher number of votes. Despite this bias, most past works [2], [3] have developed techniques using this score to classify a review as helpful or not or predict the score of a review. Most of these methods do not produce personalized recommendations, and most importantly, they do not consider the user's initial knowledge about the product. Moreover, their search space is limited only to the target product's reviews, not considering reviews of similar items sharing various aspects.

In this paper, we work on the problem of recommending item reviews to a user that, after reading a set of reviews about an item, she still needs *new information* to increase her knowledge about that item to make a better purchasing decision. We refer to an item as any product or service offered by an e-commerce website. This problem is significant since studies [4] have shown that most users only read reviews from the first page of results, and if the most "helpful" reviews do not contain what she is looking for, then she needs to search the rest of the reviews, which could be overwhelming.

We propose ***NovRev***, a recommender system of **Nov**el **Rev**iews. *NovRev* is a new *unsupervised* approach to recommend a personalized subset of unread reviews that can increase a user's knowledge about a product on different and novel aspects. It is unsupervised because we do not use the helpfulness score in the recommendation process to avoid the mentioned issues with this score.

As most users do not look past the first page of results [4], the information initially presented to a user must be helpful. Most e-commerce websites have mechanisms to show users those reviews considered most valuable. We call these initial reviews the *user's reviewing context*. This context gives a user a general idea about the product characteristics and other users' feelings towards it [5]. As far as we know, previous works have not explored the use of this context.

Considering this context, *NovRev* analyzes each candidate review to evaluate the novelty and relevance of its content. We can describe a product using a set of aspects (e.g., duration of batteries, safety), and users can refer to these by different terms. *NovRev* gives more importance to those reviews with higher aspect coverage and whose aspect terms are more relevant and novel (see Section IV-B).

In all, we summarize our contributions:

- We present ***NovRev***, a novel unsupervised approach, to recommend item reviews to users looking to increase their knowledge about an item of interest.
- We propose an approach to form a set of personalized candidate reviews. These candidate reviews consider the

This work has been supported by the National Council of Science and Technology of Mexico (CONACYT) #602434/440684, and National Science Foundation of USA (NSF) #1633330, #1345232, and #1757207

[1]"The 2019 amazon consumer behavior report"

[2]https://www.tripadvisor.com/TripAdvisorInsights/w733

type and novelty of users' reviews and their writing style.
- We present a method to use the *user's reviewing context* to find a set of reviews to recommend to users.
- We leverage a scoring function, considering the reviews' aspect coverage and the novelty and relevance of their aspect terms.
- We conduct extensive experiments on real-life datasets from different product domains.

This paper is organized as follows. Related works are reviewed in Section II. In Section III, we define the problem of recommending new reviews to users. We explain our proposed solution in Section IV. Section V presents our experimental evaluation and results. The conclusions are presented in Section VI.

II. RELATED WORK

A. Review helpfulness

The review helpfulness score indicates to the users of an e-commerce website whether a product review provides useful information for buying decisions. Past works have used different features to classify a product review as helpful or not. Regarding this score, studies [6] have shown that it is not a *good* indicator of the actual helpfulness of reviews. These studies found that users tend to give higher votes to those reviews showing more positive sentiments towards the product. Similarly, the larger the review's text, the more votes it receives, even if it does not necessarily contain the most helpful information. Another factor is the case of *"the rich get richer"* [7] for those early reviews rated as helpful. Moreover, the writer's local popularity also affects how readers perceive a review's helpfulness [3]. Therefore, a method not relying on this helpfulness score is needed to avoid biased results.

Current methods do not consider the user reviewing context, which is essential because it helps the user learn the item's fundamental characteristics and how other users perceive it [8]. *NovRev* is different because it does not rely on the helpfulness score to get the most helpful reviews for an active user. To a large extent, *NovRev* exploits the reviews initially presented to users by the e-commerce website as a knowledge context. Then, it looks for those candidate reviews that maximize the user's initial knowledge about the item. Moreover, *NovRev* also analyses reviews from similar items to find those novel and relevant aspects terms better. Most importantly, the recommended reviews by *NovRev* are personalized for each user.

B. Novelty detection

Novelty detection is an important topic in modern text retrieval systems. Previous works on this domain define similarity metrics to compare each new document with a set of previously seen documents. If they are sufficiently different, they are considered novel. *Maximal Marginal Relevance* (*MMR*) [9] and *BM25* [10] use a user query and similarity functions to rank documents. *MMR* is a summarization technique to capture relevant but not redundant information combining novelty and relevance. *BM25* is a probabilistic model that incorporates document attributes to find the most relevant document. For reviews, *RevRank* [11] uses a lexicon to define feature vector representations for each review. These representations are then compared with a "virtual" representation of what a useful review must look like to find those more similar, and consequently, the most helpful reviews.

Unlike these works, for a candidate review, *NovRev* analyzes its novelty on topics and aspect terms within context reviews and reviews for similar items within the same category. To the best of our knowledge, using reviews of items from the same category have not been explored before.

III. PROBLEM DEFINITION

An item e can have hundreds or even thousands of reviews. Let D represent all the reviews for an item, where r is a review in D. E-commerce websites generally group items based on their aspects into different categories. Let R be the set of reviews for all the items of the same category. Some e-commerce websites help users by presenting to them informative and helpful reviews [1]. For a review, its helpfulness score is calculated based on the votes given to it by users. Unfortunately, a large number of reviews have less than five votes or none, making it hard to identify their helpfulness [12]. Consequently, reviews with interesting information but with a low number of votes might not be shown to users.

A. Context reviews, candidate novel reviews, and aspect terms

For this problem, we consider the case of a user who finishes reading a set of reviews about an item but wants to read additional reviews to get more information regarding the item to make a better purchase decision. These reviews with new information could be among hundreds or thousands of other reviews, with a few or none helpful votes. In this scenario, we define two concepts: *Context Reviews* and *Candidate Novel Reviews*.

Definition 1 (Context Reviews (CR)): This is the set of reviews about an item initially presented to a user by the e-commerce website or initially chosen to be read by this user. Formally, $CR = \{r_i | r_i \in D\}$ and $CR \subset D$.

We assume a user reads these initial reviews, as established in different studies [4], [13], and the information presented by *CR* gives a user a general idea about the item.

Definition 2 (Candidate Novel Reviews (CNR)): This is the set of an item's reviews that a user has not read. Formally, $CNR = \{r_j | r_j \in D \setminus CR\}$.

Reviews in *CNR* may or may not contain new information that has not been presented in *CR*. Users can express their opinions about item aspects in their reviews. In this work, we use *Topic* and *Aspect* interchangeably.

Definition 3 (Aspect): An aspect (or topic) is a feature (i.e., characteristic) of an item on which users express their opinion.

Items belonging to the same category usually share some aspects (e.g., the sound quality of headphones). We denote the set of aspects for the items in the same category C as $A(R)$, the set of aspects in the item's reviews as $A(D)$, and aspects in a single review as $A(r)$. Users express their own opinion about an item's aspect using different terms. We are

only interested in terms related to the item's aspects.

Definition 4 (Aspect term): This is a word or combination of words that explicitly or implicitly refers to an item's aspect.

We denote the set of all aspect terms in a review as $AT(r)$, the set of aspect terms in ***CR***, and ***CNR*** as $AT(CR)$ and $AT(CNR)$, respectively. The set of aspect terms for all items in the same category C is $AT(R)$.

Some aspect terms in CNR are also in $AT(CR)$. The set of *unknown aspect terms* of an item is the set of this item's aspect terms appearing only in CNR but not in CR. I.e., $AT_{Unknown} = \{a_i | a_i \in AT(CNR) \setminus AT(CR)\}$ where a_i is an aspect term.

B. Novel reviews

Knowledge is any previously unknown and potentially useful information according to a user's criteria [14]. Recommending reviews with new and useful information to users is the main target of our work. The novelty concept follows the definition of a *new idea* [15], where a *new idea is a piece of text consisting of known and unknown terms within the same context in the text*. Leveraging this definition, we define the concept of a *novel review*.

Definition 5 (Novel Review): A review $r \in CNR$ is a novel review when the item's *unknown* aspect terms ($AT_{Unknown}$) add new *information* to the user considering the item's *known* aspect terms in CR.

If we stick to this definition, any review in CNR with a single *unknown* aspect term could be recommended to the user. To avoid this issue, we also consider the novelty degree of a review by the *coverage* and *relevance* of new information in the item's category.

Definition 6 (Review Coverage): The coverage of a review r denoted as $coverage(r)$ is the percentage of aspects existing within an item's review, given the total possible aspects from an item's category.

We represent the coverage of a review as $coverage(r) = \frac{|A(r)|}{|A(R)|}$. The higher the coverage of a review, the more information that this review presents. Not all aspects terms from different aspects increase this information in the same proportion. Thus, we introduce the *probability of an aspect term* to capture the usefulness of a review given the ***CR***.

Definition 7 (Probability of an aspect term): Given an item e, the set of reviews R for items in e's category and a candidate novel review set CNR for e, the probability of an aspect term $a \in AT(CNR)$ is defined to be

$$prob(a) = \frac{count(a, AT(CNR)))}{count(a, AT(R))} \quad (1)$$

where $count(a, AT(CNR))$ and $count(a, AT(R))$ are the frequencies of the aspect term a in CNR and R, respectively. Theoretically, $prob(a)$ is in the range of (0,1] because of $AT(CNR) \subseteq AT(R)$. However, the highest probability value 1 cannot be achieved in most situations because $|AT(CNR)| \ll |AT(R)|$ where $count(a, AT(R))$ considers the occurrences of the aspect term a within e's reviews and in the reviews of similar items within the same category of e. This step can help decrease the relevance of noisy aspect terms (e.g., synonyms). Past works only analyze the reviews of item e, but not the reviews of items in e's same category. The probability of an aspect term indicates how much it contributes to the increase of knowledge for readers and how it contributes to reviews' usefulness. The aspect terms that are more frequently used in reviews (mainly when they occur in CR) carry less novel knowledge; therefore, they should have lower values.

Overall, we are interested in those candidate reviews with high coverage of aspects and where the probability of the aspect terms for those aspects is also high.

Definition 8 (Problem Statement.): Given a set of reviews D ($D \subseteq R$) about an item, a set of context reviews CR, and a scoring function $score(\cdot)$ (see Section IV-B3), the **problem** is to find a set of reviews $N \subset CNR$, such that N maximizes the knowledge of a user about the item. Formally, we want to present to a user a set of reviews $N = \{r_1, r_2, \cdots r_K\}$, such that $\forall r_k \in N, \nexists r_l \in D \setminus N$ s.t. $score(r_l) > score(r_k)$ (i.e., the returned set contains the top-K reviews with the best scores.)

IV. PROPOSED APPROACH

NovRev consists of two phases: (i) extraction of personalized candidate reviews based on active user preferences and (ii) recommendation of novel reviews.

A. Extraction of personalized candidate reviews

To recommend personalized novel reviews, we exploit previous studies about social context on e-commerce [16]. We consider that a user likes those reviews written in a similar way that this user writes her reviews. We can have an idea of these reviews by examining the products purchased and reviewed by this user. We want to find different characteristics of a user's writing style. We use three metrics on the users' reviews to quantify a user profile: *(i) the type of content*, *(ii) the writing style*, and *(iii) the novelty of the content*.

1) Obtaining aspect terms

We find aspect terms through the identification of noun phrases from all the reviews. We run a grammar rule and Part-of-Speech (POS) [17] tagging algorithm to get noun phrases whose grammatical category is related to determiners, adjectives, and nouns (i.e., DT, JJ*, NN*). After this step, we obtain aspect terms such as *"dj style headphones"* and *"easy installation."*

Unlike other works, we are not interested in the polarity of the aspect terms. We do this to recommend reviews for the information they contain, regardless of the sentiment they express. In this way, we could recommend either positive or negative reviews to the users. Moreover, we want to avoid the likely bias towards sentiments, as mentioned in Section II-A.

2) Type of content

Users have different levels of expertise about the products. Studies [18] have shown that when *expert* users *read* reviews, they look for information that helps them build a "bank of knowledge." On the contrary, *novice* users or *regular* consumers only read reviews to have more information to better

support a purchase decision. This behavior is also present in how these users *write* reviews [19]. Therefore the types of reviews a user like to read and the type of reviews she writes are more likely similar. With this in mind, and based on previous studies [16] on e-commerce, we assume that, if a user writes very detailed reviews (e.g., reviews containing several aspect terms), the user is more likely to prefer to read detailed reviews. If a user writes reviews with fewer details (e.g., one or two aspect terms), the user would prefer similar simple reviews. Our solution should then recommend those reviews with a similar number of aspect terms as the user likes.

First, we find the reviews written by the user for items in categories different from the target item's category. Second, for each category, we calculate the average number of aspect terms. We get two vectors, denoted as *mean-aspect-term-frequency* vector, one for the given user and one for all the other users. We compare these two vectors using their geometric mean. We use the geometric mean, given it is the proper mechanism to evaluate averages on data [20]. This comparison shows if the active user uses more or fewer aspect terms than the users' average. We get the number of aspect terms in each review of CNR. Then, reviews whose number of aspect terms match the active user's aspect term profile are considered candidates. We call this set of candidate reviews R_{num}.

3) Writing style

Another essential piece of information is the user's writing style. It reflects whether the user writes reviews with a high or a low complexity level. This information is meaningful because reviewers can transmit their expertise by employing specialized terminology about the relevant topic [21]. Based on the active user's writing level, our solution should recommend those reviews displaying a similar readability level. To carry out this task, we rely on the *Automated Readability Index (ARI)* [22], a readability test designed to evaluate the intricacy of a text. ARI outputs a number that approximates a person's age and grade level needed to comprehend a text. Using ARI, we can formulate an active user's profile covering her writing style. ARI is defined as:

$$ARI = 4.71(letters/word) + 0.5(words/sentence) - 21.43 \quad (2)$$

First, we form the *mean-readability* vector for all users containing the average readability scores of each category in which the active user has written a review. Second, for the active user, we also calculate a *mean-readability* vector with the readability scores for all the categories from this user's reviews. We compare the geometric means of these two vectors to get the readability profile (above or below average). Third, we keep those reviews from CNR matching the user's profile. We call this set of candidate reviews R_{read}.

4) Novelty of the content

According to [23], novelty is the *opposite* of popularity. With this in mind, we need to find candidate reviews where the level of "popularity" of their aspect terms is the opposite of the popularity preferences of the aspect terms written by the active user. For the categories in which the active user has written a review, we get two *mean-popularity* vectors, one for all the users and one for the active user. First, for each category, we find the *top-P* frequent aspect terms. For each review, we get the percentage of aspect terms within the *top-P*. Then, we compute each category's average and compare both vectors' geometric mean to determine the user's novelty profile. Finally, for CNR, we get the *top-P* aspect terms, and for each candidate review, we determine its percentage of aspect terms inside this *top-P*. We consider as candidates those reviews whose popularity percentage is lower than the active user's popularity percentage. We denote the set of reviews obtained by considering the novelty factor as R_{inn}.

5) Candidate novel reviews set

The previous steps give us the set of candidate reviews R_{num} by using the type of content, the candidate review set R_{read} by considering the writing style, and the candidate review set R_{inn} by considering the novelty of the content. The personalized set of *Candidate Novel Reviews* for an active user is $CNR_u = R_{num} \cup R_{read} \cup R_{inn}$. If the active user has not written any reviews, then $CNR_u = CNR$.

B. Recommendation of novel reviews.

Recall that we have the set of *Context Reviews* (CR) and the personalized candidate novel reviews set CNR_u. In this section, we explain the steps to generate recommendations.

1) Calculation of the review contribution

Helpful reviews cover multiple aspects of an item [24], and different aspect terms belong to one aspect. Given the target item's domain, we are interested in the aspects (or topics) mentioned in its reviews covering as much information as possible. Using *Latent Dirichlet Allocation (LDA)* [25], we extract *K topics* from the item category reviews. For each review in CR, we find the topics for each of their sentences. Then, we create a *review topic profile* with the topics covered in each sentence. For example, we can represent these topic profiles, as shown in Table I. Each cell represents the number of sentences belonging to a particular topic for each review. The union of these topics forms a set of topics covered by CR. Following the example, the $Topics_{context}$ for CR is {*Topic1, Topic2, Topic3, Topic4, Topic5*}. Then, by the coverage Definition 6, we say that the reviews from the context cover 50% of all possible topics for this item (i.e., 5 out of 10). Using a similar procedure, we can extract $Topics_{candidate}$, as shown in Table I (in bold). Given the $Topics_{context}$ and the $Topics_{candidate}$, we calculate the gain in coverage as:

$$coverage_gain(c_i) = |Topics_{candidate} \setminus Topics_{context}| \quad (3)$$

From the example, if we add $candidate_review_1$ to the context reviews, the covered topics increase from 5 to 6, with a gain of 1, because the candidate review contains a new topic (i.e., *Topic 7*), not present in CR. To rank a candidate review $c_i \in CNR_u$, we define the concept of *total coverage* as

$$total_coverage(c_i) = coverage(c_i) + coverage_gain(c_i) \quad (4)$$

Here, $coverage(c_i) \in (0, 1]$ and $coverage_gain(c_i) \in [0, +\inf)$. The $coverage_gain(c_i)$ term is more significant in

TABLE I
TOPIC PROFILES (# OF OCCURRENCES OF TOPICS IN REVIEWS)

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
$context_review_1$	1	0	2	1	0	0	0	0	0	0
$context_review_2$	0	1	2	0	1	0	0	0	0	0
$context_review_3$	0	0	1	2	0	0	0	0	0	0
$candidate_review_1$	**0**	**0**	**2**	**0**	**1**	**0**	**2**	**0**	**0**	**0**

the ranking process when their values are above one. However, for two reviews with the same coverage gain, the original coverage value $coverage(c_i)$ is more significant.

We introduce another concept, *support*, to facilitate the selection of candidate reviews when several of them have the same coverage gain. Intuitively, if a candidate review's topics are more similar to the topics for reviews in $R \setminus D$, this candidate review has higher support. For a candidate review, we calculate the similarity score of its topics and the topics of reviews in $R \setminus D$. If this similarity score is higher than a given threshold σ_{sim}, we put similar reviews in $R \setminus D$ to a new set *similars*. Formally, we define the *support* of a candidate review $c_i \in CNR_u$ as follows:

$$support(c_i) = \frac{|similars|}{|R \setminus D|} \tag{5}$$

where $|\cdot|$ is the set cardinality operator. Therefore, *support* can help to break ties when multiple candidate reviews have the same coverage gain. A candidate review should increase the knowledge of a user towards an item. We quantify this contribution of knowledge from a candidate review as:

$$contribution(c_i) = total_coverage(c_i) + support(c_i). \tag{6}$$

2) Calculation of the review importance

We consider important reviews to be those with higher probability (Definition 7) and more novel (Definition 5) *unknown* aspect terms. To calculate a review's importance, we need to consider the relevance of each unknown aspect term $a_u \in AT_{Unknown}$, which is defined as follows.

$$relevance(a_u) = \alpha \cdot (-log_2(\theta_{a_u})) + (1-\alpha) \cdot \left(log_2 \left(\frac{\theta_{a_u}}{\pi_{a_u}} \right) \right) \tag{7}$$

where θ_{a_u} is the probability of a_u appearing in CNR (i.e., $\theta_{a_u} = \frac{count(a_u, AT(CNR))}{|AT(CNR)|}$), π_{a_u} is the probability of a_u occurring in reviews for the target item's category (i.e., $\pi_{a_u} = prob(a_u)$). α is a value between 0 and 1 that determines the relative importance of Eq. 7 terms. It is natural to use the probability of unknown aspect terms in an item's candidate novel reviews (captured with θ_{a_u}) to calculate this relevance score. However, using it alone is insufficient because the probability θ_{a_u} is higher for popular unknown aspect terms. The relevance definition further utilizes another term, the probability of unknown aspect terms in all the similar items' reviews (captured with π_{a_u}) to balance this score.

When evaluating a review's novelty, we also consider the time since their first publishing. Reviews with recent *unknown* aspect terms are preferred because they indicate innovations that the item currently has (i.e., their recency is an indicator of their novelty). Similarly, due to a "rich get richer" effect, more recent reviews are rarely read [26]. Therefore, we would like to downplay reviews with older aspect terms. We define the earliness of an *unknown* aspect term $a_u \in AT_{Unknown}$ as:

$$earliness(a_u) = e^{-(\frac{days(a_u)}{\lambda})} \tag{8}$$

where $days$ is the number of days elapsed since the first mention of the *unknown* aspect term in the candidate reviews, and λ is a time window (e.g., 30 days). Having the $relevance$ and the $earliness$ of $a_u \in AT_{Unknown}$, we can determine the overall $importance$ of the $unknown$ aspect terms in a candidate review c_i as follows:

$$importance(AT_{Unknown_{c_i}}, c_i) = \frac{\sum_{a_u \in AT_{Unknown_{c_i}}} relevance(a_u) \cdot earliness(a_u)}{|AT_{Unknown_{c_i}}|} + \frac{|AT_{Unknown_{c_i}}|}{|AT(c_i)|} \tag{9}$$

where $AT_{Unknown_{c_i}}$ has all the *unknown* aspect terms in c_i. The importance of a candidate review c_i, $importance(c_i)$, is determined by the unknown aspect terms in c_i. I.e., $importance(c_i)$ is the same as $importance(AT_{Unknown_{c_i}}, c_i)$.

3) Reviews recommendation.

Finally, we can score each candidate review c_i based on its contribution to the knowledge coverage and the importance of its *unknown* aspect terms with

$$score(c_i) = \beta \cdot contribution(c_i) + (1-\beta) \cdot importance(c_i) \tag{10}$$

where β is a value between 0 and 1 that determines the relative importance of the equation terms. After scoring all reviews in CNR_u, we rank them and return to the active user the top candidate reviews with higher scores in a set N.

V. EXPERIMENTAL RESULTS

A. Experimental settings

1) Datasets

We select *eight real-life datasets* from the Amazon dataset repository [27]: *Electronics* (**ELT**), *Clothing, Shoes and Jewelry* (**CSJ**), *Home and Kitchen* (**HK**), *Cell phones and Accessories* (**CEL**), *Automotive* (**AUTO**), *Toys and Games* (**TOY**), *Health and Personal Care* (**HPC**), and *Office Products* (**OP**). Each dataset contains products from several subcategories. We remove users with less than ten reviews for all datasets, reviews with less than two words, and subcategories with less than two items. Table II shows a summary of these datasets. Note that the datasets do not need to have any pre-labeled information because the new solution and the baseline methods are all unsupervised.

2) Baselines

We implement five baseline algorithms: (i) ***Random***; we randomly select N reviews from CNR. (ii) ***Helpful***, it ranks

TABLE II
STATISTICS OF DATASETS USED IN THIS PAPER

	ELT	HK	HPC	CSJ	TOY	CEL	OP	AUTO
Users	45,124	13,671	8,731	5,177	4,188	3,209	1,809	360
Items	61,900	27,265	17,878	20,341	11,524	9,012	2,283	1,579
Reviews	771,784	226,122	162,892	71,808	74,360	47,031	33,998	4,969
Subcats.	692	863	569	619	569	49	259	1,717
Words per review	∼138	∼120	∼129	∼65	∼129	∼131	∼163	∼97

the CNR by their helpfulness score and selects the top-N reviews. (iii) ***BM25*** is a popular information retrieval algorithm used for search engines. Although it is not designed to find novel reviews, it retrieves those relevant ones given an input (i.e., CR). Therefore, we would like to evaluate the returned relevant reviews' information. (iv) *Maximum Marginal Relevance* (***MMR***) iteratively selects the review that is not only relevant to CR but also dissimilar to the previously selected candidate reviews. (v) ***RevRank*** is a state-of-the-art approach for selecting the most helpful reviews. We use the same parameters as in [11]. As *NovRev*, these last three baselines are *unsupervised* approaches.

We test two variants of our approach: a) ***NovRev***, which ranks the candidate reviews based on the score function (Equation 10); and b) ***NovRevInc***, which incrementally adds the selected candidate review to the set of context reviews before finding the next candidate review to recommend.

3) Metrics

Intra-list similarity (ILS). [28]. This metric captures the similarity of the list of novel recommendations. A low ILS value represents a list with less similar items, which is important to avoid redundant information. ILS is defined as:

$$ILS_N = \frac{1}{2} \sum_{c_i \in \mathrm{N}} \sum_{c_j \in N} \mathrm{cos_sim}\left(c_i, c_j\right); c_i \neq c_j \tag{11}$$

List Novelty Cost (LNC) [29]. It captures the recommendation list cost, using the ratio of unknown aspect terms for all the aspect terms in a review. LNC is defined as follows:

$$LNC_N = \log_{10}\left(\sum_{c_i \in N}\left(1 + \gamma \cdot \left(1 - \frac{|AT_{Unknown}(c_i)|}{|AT(c_i)|}\right)\right)\right) \tag{12}$$

A review containing only *unknown* aspect terms (i.e., only new information) contributes 1 to the overall cost. On the other hand, a review with only *known* aspects term contributes to $1+\gamma$, where γ is a punishment factor. For our tests, we empirically use $\gamma = 0.4$. Low values for LNC are preferred.

Precision@k (P@k). It measures the number of relevant reviews within the first k elements in the recommended list, where $k \leq N$. We consider a candidate review relevant if the ratio between its *unknown* aspect terms and its total aspect terms is greater or equal than a given threshold. We use a value of $k = 3$ and a threshold $t = 0.7$. High values indicate that more relevant reviews contain new information.

$$relevant(c_i) = \begin{cases} True & \frac{|AT_{Unknown}(c_i)|}{|AT(c_i)|} \geq t \\ False & \text{otherwise} \end{cases} \tag{13}$$

Normalized Discounted Cumulative Gain (nDCG). It is the normalization of the DCG metric [30]. DCG penalizes if a relevant document appears low in the ranking normalized by the Ideal DCG ($IDCG$). High values are preferred. We select the relevant reviews, $relevant(c_i)$, as in Precision@k (Eq. 13). The $nDCG$ metric is given by

$$nDCG_N = \frac{DCG_N}{IDCG_N}, \text{ where } DCG_N = \sum_{i}^{|N|} \frac{2^{relevant(c_i)} - 1}{\log_2(1+i)} \tag{14, 15}$$

4) User, Item, and Context Reviews selection

For each dataset, we randomly pick 100 active users. For each user, we randomly pick one of her reviewed items as the item of interest. We provide five reviews as CR because most e-commerce websites generally present five to eight reviews. Among these five CR reviews, 60% (i.e., three reviews) have the highest helpfulness score, while the remaining 40% (i.e., two reviews) are randomly picked from the remaining reviews. We form the CR this way because around 60% of the initially presented reviews in existing e-commerce websites have the helpfulness score.

5) Other parameters

To calculate *novelty* (see Section IV-A4), we use $P = 0.20$. For the topic profiles (see Section IV-B1), we use $K = 15$. For the support (Eq. 5), we use $\sigma_{sim} = 0.7$. We use $\alpha = 0.5$ in Eq. 7. For Eq. 8, we use a time window of $\lambda = 30$ (i.e., 30 days). For the final score calculation in Eq. 10, we use $\beta = 0.7$. Finally, we return the first five reviews (i.e., $N = 5$).

B. Results

1) Evaluating the effectiveness of recommended reviews

We averaged the results for active users. The experiments were repeated three times, and the averages are reported in Table III. The best performance is in bold. Our models obtain the best results for all metrics. ILS results indicate that the elements in N are more different from each other, meaning that *NovRev* presents a higher number of relevant aspect terms not shown before. Similarly, for LNC, the results show that each recommended review contains more relevant *unknown* aspect terms, and these reviews contribute more to increase a user's knowledge.

The results on Precision@K and nDCG indicate that our solution can recommend more relevant reviews in higher-rank positions. These results confirm that our approach recommends those relevant candidate reviews that increase the knowledge about an item given a set of context reviews. Like the previous two metrics, the results obtained by these metrics are also coherent with the definitions of our methods.

TABLE III
TEST RESULTS FOR THE DIFFERENT DATASETS.
(ILS AND LNC LOWER IS BETTER, P@K AND NDCG HIGHER IS BETTER)

ELT							
	Random	Helpful	BM25	MMR	RevRank	NovRev	NovRevInc
ILS	0.0419	0.0489	0.0603	0.0685	0.0598	**0.0382**	0.0437
LNC	0.7296	0.7315	0.735	0.7521	0.734	**0.7286**	0.7318
P@k	0.68	0.68	0.6433	0.3067	0.6433	**0.73**	0.68
nDCG	0.8331	0.8527	0.8088	0.5151	0.8161	**0.8692**	0.8565
HPC							
	Random	Helpful	BM25	MMR	RevRank	NovRev	NovRevInc
ILS	0.0578	0.0593	0.0777	0.0791	0.0746	0.0545	**0.0544**
LNC	0.7338	0.7332	0.7342	0.7488	0.7345	**0.7286**	0.7312
P@k	0.6233	0.6667	0.65	0.2967	0.6333	**0.7367**	0.7133
nDCG	0.8263	0.8130	0.799	0.56	0.7999	**0.8662**	0.8539
TOY							
	Random	Helpful	BM25	MMR	RevRank	NovRev	NovRevInc
ILS	0.0714	0.0836	0.0951	0.1113	0.0732	**0.0692**	0.0821
LNC	0.7379	0.7383	0.7386	0.7472	0.7382	**0.7333**	0.7357
P@k	0.4119	0.3652	0.3819	0.1352	0.3852	**0.4819**	0.4385
nDCG	0.7647	0.7003	0.6789	0.466	0.7032	**0.8111**	0.7930
OP							
	Random	Helpful	BM25	MMR	RevRank	NovRev	NovRevInc
ILS	0.0702	0.0691	0.0873	0.0937	0.0866	**0.0685**	0.0686
LNC	0.7426	0.7418	0.7448	0.7564	0.7439	**0.7403**	0.7421
P@k	0.5593	0.6333	0.5519	0.2370	0.5963	**0.6519**	0.6444
nDCG	0.7380	0.8374	0.7574	0.4470	0.8203	0.8409	**0.8445**

HK							
	Random	Helpful	BM25	MMR	RevRank	NovRev	NovRevInc
ILS	0.05	0.051	0.067	0.077	0.068	**0.047**	**0.047**
LNC	0.7269	0.7258	0.7294	0.7418	0.7287	**0.7229**	0.7248
P@k	0.695	0.735	0.695	0.405	0.675	**0.7883**	0.7483
nDCG	0.8643	0.8902	0.8561	0.5839	0.8682	**0.9143**	0.9026
CSJ							
	Random	Helpful	BM25	MMR	RevRank	NovRev	NovRevInc
ILS	0.0382	0.0326	0.0446	0.0506	0.0392	**0.0301**	0.0338
LNC	0.6962	0.6971	0.6989	0.7077	0.6966	**0.6928**	0.6961
P@k	0.6333	0.8333	0.7	0.3	0.7667	**0.9**	0.8
nDCG	0.8983	0.8763	0.8855	0.7109	0.9056	**0.9151**	0.9050
CEL							
	Random	Helpful	BM25	MMR	RevRank	NovRev	NovRevInc
ILS	0.0596	0.0557	0.0821	0.0881	0.0699	0.0574	**0.0534**
LNC	0.7227	0.7203	0.7250	0.7359	0.7238	**0.7194**	0.7207
P@k	0.6533	0.66	0.6267	0.3167	0.6433	**0.6833**	0.6533
nDCG	0.822	0.8291	0.8219	0.5362	0.821	**0.8706**	0.8635
AUTO							
	Random	Helpful	BM25	MMR	RevRank	NovRev	NovRevInc
ILS	0.0424	0.0445	0.0546	0.0612	0.0557	0.0409	**0.0394**
LNC	0.6633	0.6643	0.6653	0.6691	0.6627	**0.6613**	0.6625
P@k	0.755	0.7017	0.735	0.585	0.8083	**0.8117**	0.7783
nDCG	0.8823	0.8652	0.9259	0.7345	0.9311	**0.9344**	0.9242

2) Effect of parameters

We test the performance of our proposed approaches by varying the parameters α (used in Eq. 7) and β (used in Eq. 10). We fix one and vary the other. α is related to the importance of the aspect terms in the candidate reviews and the category reviews, while β influences the final score. Due to space limitations, we only show some of the results in Fig. 1. When $\beta > 0.8$, the performance degrades for both novelty and relevancy. When less importance is given to the *unknown* aspect terms and less *novel unknown* aspect terms are considered, the recommendations are more similar and have more popular aspect terms. Similarly, when $\alpha \geq 0.5$, we can get better results. This result indicates that it is crucial to consider the occurrences of *unknown* aspect terms in reviews of items from the same category.

We also test the performance of our approaches for the coverage of the topics. Table IV reports the topic coverage for a product review from the CSJ dataset. For this specific example, because *NovRevInc* incrementally adjusts the topics covered in the current context, it can recommend those reviews with a topic coverage of 100%.

3) Effect of the threshold t

For $p@k$ and *nDCG* metrics, the hyperparameter t is used to select which reviews are considered relevant from the recommended list N. We compare *NovRev* with the best baseline for the TOY dataset. The results reported in Fig. 2 show that when t increases, both $p@k$, and *nDCG* values decrease because fewer reviews meet the relevancy condition. The results also show that our approach consistently outperforms the baseline. This result entails that reviews with new information (*NovRev*) in a higher ranking position are better than randomly picking reviews (baseline).

C. Preliminary user study

We further investigate the usefulness of our method through a user study. We want to examine whether the reviews recommended by *NovRev* are considered as carrying more information (or more relevant). We chose three datasets, TOY, OP, and AUTO, for this study because their items are most likely to be understood by most people. For each dataset, we compared the recommended reviews returned by our *NovRev* and the best performing baselines, which are *Random*, *Helpful*, and *RevRank* for TOY, OP, and AUTO, respectively. We randomly chose ten products for each dataset, generated context reviews, and recommended reviews generated by *NovRev* and the other best baseline. In total, we generated reviews for 30 products (from the three datasets). We recruited 18 users from different backgrounds having at least a bachelor's degree. We split these 30 products into six groups (i.e., five products per group) and randomly assigned three users to each group. Each user was asked to read the context reviews for every product and give feedback on which recommended reviews are helpful to increase their knowledge about the product. Overall, 61% of the participants selected the recommendation from *NovRev* (56.7% in TOY, 60% in OP, and 59.7% in AUTO). Although this study shows promising results for *NovRev*, it has one limitation. Recall that *NovRev* makes personalized recommendations using historical user reviews. However, such information is not available for the recruited users. Therefore, the reviews are not personalized to each participant. We left a more in-depth study for future research.

VI. CONCLUSIONS

This paper studied the problem of recommending personalized product reviews to those users seeking new, important, and relevant information to increase their knowledge of a product after reading previous reviews. We proposed ***NovRev***, a novel *unsupervised* approach to solve this problem. Our extensive experiments on eight real-life datasets for different item domains show that our approach presents better results than the baselines on different metrics. Our preliminary user study also provides positive evidence of the helpfulness of the

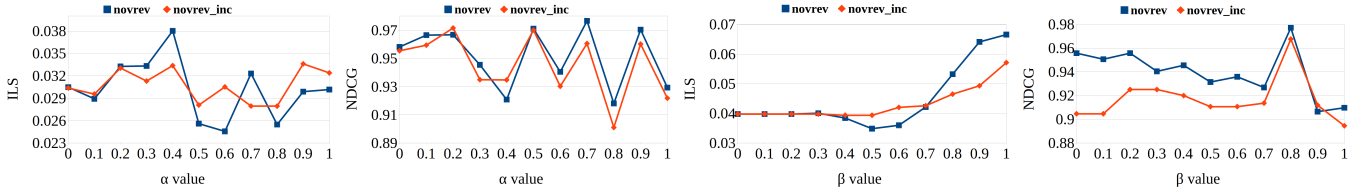

Fig. 1. Metrics comparison for the OP dataset.

TABLE IV
TOPICS COVERED FOR A PRODUCT REVIEW IN CSJ DATASET.
(CR COVERS 10 OF 15 TOPICS: {0,1,2,5,6,7,10,11,12,14})

	Random	Helpful	BM25	MMR	RevRank	**NovRev**	**NovRevInc**
New Topics Covered	{9,13}	{3,8,9}	{3,9,13}	{3,9,13}	{8,9,13}	{4,8,9,13}	{3,4,8,9,13}
Topics coverage in total (context + recommendations)	0.8	0.866	0.866	0.866	0.866	0.933	1.0

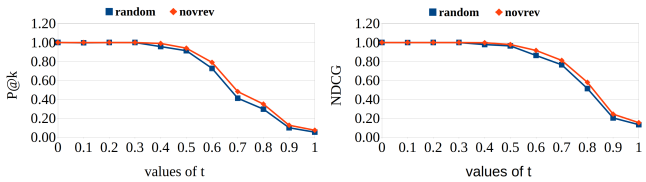

Fig. 2. Effect of threshold t in the TOY dataset.

recommended reviews.

REFERENCES

[1] G. O. Diaz and V. Ng, "Modeling and prediction of online product review helpfulness: A survey," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 698–708.

[2] Y.-J. Park, "Predicting the helpfulness of online customer reviews across different product types," *Sustainability*, vol. 10, no. 6, p. 1735, 2018.

[3] O. Mokryn, "The opinions of a few: A cross-platform study quantifying usefulness of reviews," *Online Social Networks and Media*, vol. 18, p. 100080, 2020.

[4] J. Otterbacher, "'helpfulness' in online communities: a measure of message quality," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2009, pp. 955–964.

[5] E. E. K. Kim, A. S. Mattila, and S. Baloglu, "Effects of gender and expertise on consumers' motivation to read online hotel reviews," *Cornell Hospitality Quarterly*, vol. 52, no. 4, pp. 399–406, 2011.

[6] R. Dong, M. Schaal, M. P. O'Mahony, and B. Smyth, "Topic extraction from online reviews for classification and recommendation," in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[7] E. Gilbert and K. Karahalios, "Understanding deja reviewers," in *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, 2010, pp. 225–228.

[8] J. Burton and M. Khammash, "Why do people read reviews posted on consumer-opinion portals?" *Journal of Marketing Management*, vol. 26, no. 3-4, pp. 230–255, 2010.

[9] J. G. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries." in *SIGIR*, vol. 98, 1998, pp. 335–336.

[10] S. Robertson, H. Zaragoza *et al.*, "The probabilistic relevance framework: Bm25 and beyond," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[11] O. Tsur and A. Rappoport, "Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews," in *Third International AAAI Conference on Weblogs and Social Media*, 2009.

[12] J. Li and L. Zhan, "Online persuasion: How the written word drives wom: Evidence from consumer-generated product reviews," *Journal of Advertising Research*, vol. 51, no. 1, pp. 239–257, 2011.

[13] Feedvisor, "The 2019 amazon consumer behavior report," 2019.

[14] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus, "Knowledge discovery in databases: An overview," *AI magazine*, vol. 13, no. 3, pp. 57–57, 1992.

[15] M. Alksher, A. Azman, R. Yaakob, E. M. Alshari, A. K. Rabiah, and A. Mohamed, "Effective idea mining technique based on modeling lexical semantic," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 16, pp. 5350–5362, 2018.

[16] N. Korfiatis, E. GarcíA-Bariocanal, and S. SáNchez-Alonso, "Evaluating content quality and helpfulness of online product reviews: The interplay of review helpfulness vs. review content," *Electronic Commerce Research and Applications*, vol. 11, no. 3, pp. 205–217, 2012.

[17] K. I. Muhammad, A. Lawlor, and B. Smyth, "A live-user study of opinionated explanations for recommender systems," in *Proceedings of the 21st International Conference on Intelligent User Interfaces*, 2016, pp. 256–260.

[18] Y. Kang and L. Zhou, "Longer is better? a case study of product review helpfulness prediction," 2016.

[19] D. Plotkina and A. Munzel, "Delight the experts, but never dissatisfy your customers! a multi-category study on the effects of online review source on intention to buy a new product," *Journal of Retailing and Consumer Services*, vol. 29, pp. 1–11, 2016.

[20] H. Mangesius, D. Xue, and S. Hirche, "Consensus driven by the geometric mean," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 251–261, 2016.

[21] M. Li, L. Huang, C.-H. Tan, and K.-K. Wei, "Helpfulness of online product reviews as seen by consumers: Source and content features," *International Journal of Electronic Commerce*, vol. 17, no. 4, pp. 101–136, 2013.

[22] R. Senter and E. A. Smith, "Automated readability index," CINCINNATI UNIV OH, Tech. Rep., 1967.

[23] M. Mendoza and N. Torres, "Evaluating content novelty in recommender systems," *Journal of Intelligent Information Systems*, pp. 1–20, 2019.

[24] Y. Yang, C. Chen, and F. S. Bao, "Aspect-based helpfulness prediction for online product reviews," in *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2016, pp. 836–843.

[25] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[26] S. Moghaddam, M. Jamali, and M. Ester, "Etf: extended tensor factorization model for personalizing prediction of review helpfulness," in *Proceedings of the fifth ACM international conference on Web search and data mining*, 2012, pp. 163–172.

[27] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *proceedings of the 25th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2016, pp. 507–517.

[28] T. Murakami, K. Mori, and R. Orihara, "Metrics for evaluating the serendipity of recommendation lists," in *Annual conference of the Japanese society for artificial intelligence*. Springer, 2007, pp. 40–46.

[29] A. Omari, D. Carmel, O. Rokhlenko, and I. Szpektor, "Novelty based ranking of human answers for community questions," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 215–224.

[30] Y. Wang, L. Wang, Y. Li, D. He, W. Chen, and T.-Y. Liu, "A theoretical analysis of ndcg ranking measures," in *Proceedings of the 26th annual conference on learning theory (COLT 2013)*, vol. 8, 2013, p. 6.