Selecting Comparative Sets of Reviews Across Multiple Items

Trung-Hoang Le
Singapore Management University
Singapore
thle.2017@smu.edu.sg

Hady W. Lauw
Singapore Management University
Singapore
hadywlauw@smu.edu.sg

ABSTRACT

While choosing among several products, users may look up reviews from each product they are considering. Due to the large number of reviews of products, selecting representative reviews from one product alone is already a challenging problem. In this work, we further aim to conduct review selection for multiple products simultaneously for comparative purposes. We formulate objective functions that synchronize the review selection and design efficient algorithms to optimize for the objective functions. To narrow down the potentially long list of comparison items into a shorter list of more similar items, we construct a graph representing items' similarity and design efficient algorithms to find the heaviest k-subgraph including the target item. The results are validated on real world datasets on various product categories.

1 INTRODUCTION

E-commerce is now the predominant means for procuring items. Unlike brick-and-mortar stores, e-commerce sites are not severely limited by inventory shelf spaces and can offer many options for every consumer intent. Given the large number of alternatives to consider, and little prior experience with them, consumers resort to product reviews to glean as much information as they can from the experiences of others as documented in the reviews.

Reviews are so much a part of the e-commerce landscape now that virtually every store features reviews. So much so, that nowadays it is common to find products with thousands of reviews, if not more. What was originally a mechanism to address the paradox of choice – of which products to purchase – has now itself turned into another paradox of choice – of which reviews to read.

If a consumer only has time to read a few reviews, which among the many (potentially thousands) should they read? In the literature, this question has been looked into from multiple angles (see section 5). One option is to let users vote on which reviews have been helpful, but this may not give a fair chance to all reviews as even the voters may have only seen a few reviews. Another option is to create a summary of all the reviews, but this summary, either being crafted by a machine learning model or assembled from many reviews, may not have the original authenticity of a genuine review.

This work follows the line of research in selecting a small number k of reviews that are "representative" of the full set of reviews of a given product. There are various ways to define representativeness as surveyed in section 5. One that is particularly relevant is *characteristic* review selection [15], which seeks to find a subset of reviews that collectively cover both positive and negative opinions of product aspects in a proportion that is close to the overall. Intuitively, by reading the few selected reviews,

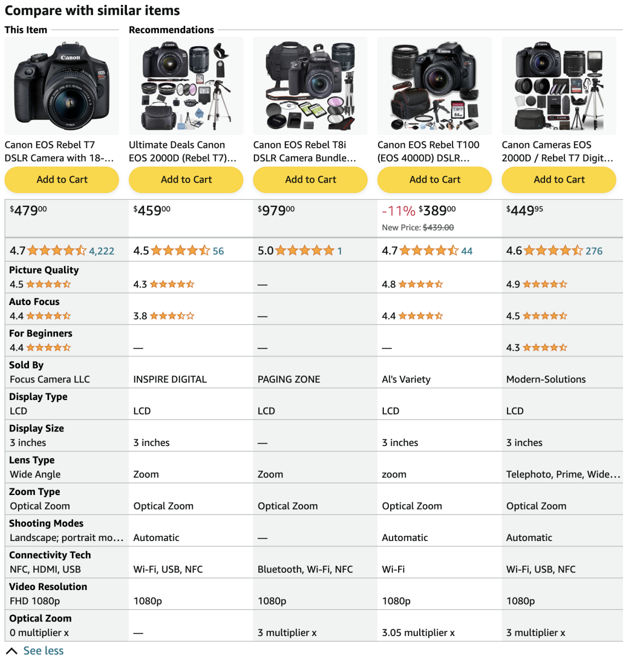


Figure 1: An example of "Compare with similar items" (This is captured on Amazon.com)

a consumer would be well-versed in considering the trade-offs associated with a product.

Comparative Review Selection. Notably, in the existing literature, review selection is conducted for an *individual* product independently. In this work, we are interested in selecting comparative reviews from *multiple* products simultaneously. We posit that a consumer's decision making is not simply binary in the sense of whether to purchase a product. Rather, it is usually comparative in the sense of which among a few alternatives to decide upon. For instance, on certain e-commerce sites such as Amazon.com, when consumers are viewing a target item (e.g., Canon EOS Rebel T7 DSLR Camera[1]), they may be presented with a number of "similar" items, ostensibly due to similarity in attributes or specifications, as illustrated in Figure 1. There are yet other means of identifying comparative items such as also bought items, also viewed items.

As shown in Figure 1, each item could have hundreds to thousands of reviews. Beyond the hard specs, consumers would likely still wish to read the reviews. Given a target product and a number of comparative products, our primary focus in this work is on selecting reviews from the given products in such a way that the selected reviews would be representative of the respective products, and simultaneously *the selected reviews would cover similar aspects that would facilitate comparison across those products.* This latter objective is novel to this work. It also gives rise to a new

[1] https://www.amazon.com/Canon-Rebel-T7-18-55mm-II/dp/B07C2Z21X5

problem formulation as what used to be a combinatorial selection across reviews of one product now becomes combinatorial explosion across multiple products. We formulate synchronized review selection objectives and propose algorithms towards approximating them.

While a consumer is then presented only with a small number of reviews, reading a few reviews across multiple products could still be taxing on the mind. Thus, to further ease the cognitive load on consumers, as a secondary objective, we would build on the aforementioned review selection objective to narrow down the given (long) list of comparative products to a smaller sized list of core comparative products. By formulating the products or items as a graph of vertices, with the edge weights reflecting the similarities across their selected reviews, we turn the problem into finding the heaviest weight top-k items including the target item.

Contributions. We make several contributions in this work. First, we propose a novel review selection problem for selecting comparative review sets for multiple items. Second, we formulate the objective function that synchronizes the review selection process and design efficient algorithms to solve this objective function. Third, we describe an efficient heuristic approximation to find top-k similar items among the candidate comparative items that must include the target item. Fourth, we conduct experiments on real world data to validate the efficacies of the proposed algorithms against comparable baselines. Finally, we make our code publicly available[2] for reproducibility.

Novelty. The problem in selecting comparative sets of reviews across multiple items is novel, which is distinct from previous work sorely focus on selecting a set of reviews for a single item. This necessitates a new solution, which we formulate based on integer regression. Moreover, the subsequent task of narrowing down the core list of comparative items is also novel, which is formulated as selecting the heaviest k-subgraph including the target item to maximize the similarity among these selected k items, differing from the existing works in selecting the heaviest k-subgraph. As the problem is intractable, a heuristic approximation algorithm is proposed.

2 COMPARATIVE REVIEW SETS SELECTION ACROSS MULTIPLE ITEMS

We first present our problem formulations, then describe the proposed algorithm to approximate the otherwise intractable problem. Table 1 lists the main notations used in this paper.

2.1 Problem Formulations

2.1.1 Comparative Review Sets Selection. Given a collection of n items $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, each item p_i has a collection of reviews $\mathcal{R}_i$ discussing aspects from a universal set of z aspects $\mathcal{A} = \{a_1, a_2, \ldots, a_z\}$. A typical review only comments on a subset of these aspects, expressing positive or negative opinions. Given a collection of reviews $\mathcal{S}_i \subseteq \mathcal{R}_i$, we use $\pi(\mathcal{S}_i) \in \mathbb{R}_+^{2z}$ to denote the opinion vector that represents the distribution of opinions of $\mathcal{S}_i$. Throughout the paper, we assume that aspects and opinions can be extracted automatically, e.g., using frequency-based approach [5] or Sentires [42], and we consider them as given.

Let $\phi(\mathcal{S}_i) \in \mathbb{R}_+^z$ be the vector representing aspect distribution of $\mathcal{S}_i$ (just the aspects, irrespective the opinion of individual items). When comparing two items, we often base on common aspects of both items regardless of their opinions to see how they are different from each other. For every item, we would like to select a subset of reviews that characterize the item well. We also want the selected sets of reviews to be similar to one another, e.g., discussing same aspects, so we can compare more directly.

[2]https://github.com/PreferredAI/CompaReSetS

Table 1: Main Notations

Symbol	Description
$\mathcal{P}$	set of n products $\{p_1, p_2, \ldots, p_n\}$
$\mathcal{A}$	set of z aspects $\{a_1, a_2, \ldots, a_z\}$
$\mathcal{R}_i$	set of all reviews of item p_i
$\mathcal{S}_i \subseteq \mathcal{R}_i$	a subset reviews of $\mathcal{R}_i$
m	maximum number of reviews to be selected
k	top-k most similar items to be selected
$\pi(\mathcal{S}_i)$	opinion distribution vector of $\mathcal{S}_i$
$\phi(\mathcal{S}_i)$	aspect distribution vector of $\mathcal{S}_i$
τ_i	target opinion distribution vector for item p_i
Γ	target aspect distribution vector
$\Delta(x, y)$	distance of two vector x and y, i.e., L^2 distance
λ	control factor of opinion over aspect
μ	control factor of comparisons among items
$[a; b]$	concatenation of vector a and b
ω, ν	indicators of which reviews to be selected
$\rho \subseteq \mathcal{P}$	a solution set of products selected by TargetHkS
γ_i	indicator of whether item p_i is a part of solution ρ
w_{ij}	the similarity between two nodes p_i and p_j

Specifically, for any two items p_i and p_j, the selected sets are $\mathcal{S}_i$ and $\mathcal{S}_j$ respectively, we would like to minimize the distance between two aspect distribution vectors $\phi(\mathcal{S}_i)$ and $\phi(\mathcal{S}_j)$. We also use the notion of a target aspect vector Γ, acting as an independent optimization goal, i.e., aspect vector of the target item. The formal problem formulation is as follows:

Problem 1. Comparative Review Sets Selection (CompaReSetS). *We are given a collection of n items $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, where p_1 is regarded as the target item and p_2 to p_n as comparative items. Every item p_i has a collection of reviews $\mathcal{R}_i$, a target opinion vector τ_i. For a target aspect vector Γ and an integer number m, find $\mathcal{S}_i \subseteq \mathcal{R}_i$ for every item p_i such that $|\mathcal{S}_i| \leq m$ and*

$$\sum_{i=1}^{n} \Delta(\tau_i, \pi(\mathcal{S}_i)) + \lambda^2 \sum_{i=1}^{n} \Delta(\Gamma, \phi(\mathcal{S}_i)) \quad (1)$$

is minimized.

Where m is the maximum number of reviews to be selected, $\tau_i = \pi(\mathcal{R}_i)$ is the target opinion distribution vector, $\lambda \geq 0$ is the tradeoff factor between distance of opinion vectors and distance of aspect vectors, and $\Delta(x, y)$ is the distance between two vectors $x \in \mathbb{R}^l$ and $y \in \mathbb{R}^l$, i.e., squared Euclidean distance,

$$\Delta(x, y) = (x - y)^2 = \sum_{i=1}^{l} (x_i - y_i)^2 \quad (2)$$

Equation 1 can be solved separately for each item p_i, minimizes:

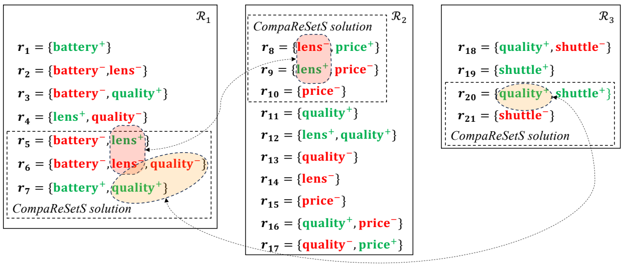
$$\Delta(\tau_i, \pi(\mathcal{S}_i)) + \lambda^2 \Delta(\Gamma, \phi(\mathcal{S}_i)) \quad (3)$$

Based on the distance metric in Equation 2, we can rewrite Equation 3 as follows:

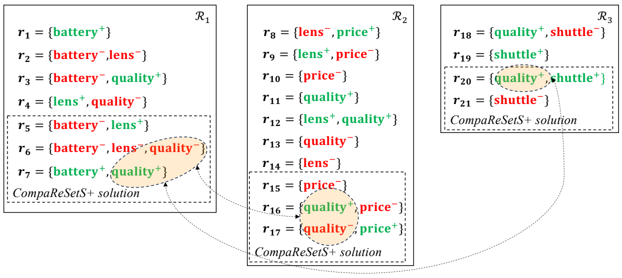
$$\Delta([\tau_i; \lambda \cdot \Gamma], [\pi(\mathcal{S}_i); \lambda \cdot \phi(\mathcal{S}_i)]) \quad (4)$$

Where the target vector $[\tau_i; \lambda \cdot \Gamma]$ is simply the concatenation between τ_i and $\lambda \cdot \Gamma$.

Working example 1. Figure 2a shows an example of 3 items $\{p_1, p_2, p_3\}$ and a solution of CompaReSetS. The frequencies of aspects {battery, lens, quality, price, shuttle} in $\mathcal{R}_1$ are $\{6, 4, 4, 0, 0\}$. With opinions {battery$^+$, battery$^-$, lens$^+$, lens$^-$, quality$^+$, quality$^-$, price$^+$, price$^-$, shuttle$^+$, shuttle$^-$}, the target opinion vector would

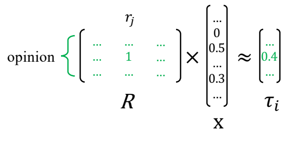


(a) CompaReSetS solution

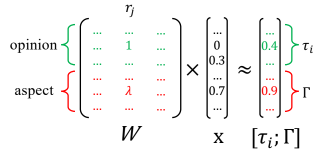


(b) CompaReSetS+ solution

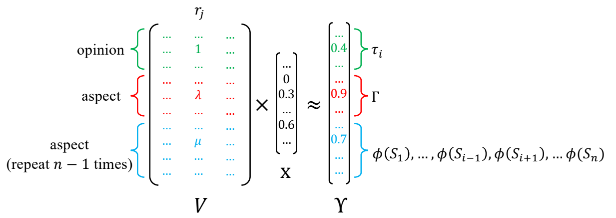
Figure 2: An example of selecting comparative review sets across 3 items $\{p_1, p_2, p_3\}$



(a) Crs



(b) CompaReSetS



(c) CompaReSetS+

Figure 3: Linear regression visualization on item p_i

be $\tau_1 = \pi(\mathcal{R}_1) = \left(\frac{2}{6}, \frac{4}{6}, \frac{2}{6}, \frac{2}{6}, \frac{2}{6}, \frac{2}{6}, 0, 0, 0, 0\right)$, where the denominator 6 is the maximum occurrences of aspects. The target aspect vector is $\Gamma = \phi(\mathcal{R}_1) = \left(\frac{6}{6}, \frac{4}{6}, \frac{4}{6}, 0, 0\right)$. With $\lambda = 1$, presuming a selection of $m \leq 3$ reviews describing τ_1 and Γ, the optimal set of $\mathcal{R}_1$ is $\mathcal{S}_1 = \{r_5, r_6, r_7\}$ as it has the identical opinion vector $\pi(\mathcal{S}_1) = \left(\frac{1}{3}, \frac{2}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0, 0\right) \equiv \tau_1$ and identical aspect vector $\phi(\mathcal{S}_1) = \left(\frac{3}{3}, \frac{2}{3}, \frac{2}{3}, 0, 0\right) \equiv \Gamma$. Similarly, the optimal sets of $\mathcal{R}_2$ and $\mathcal{R}_3$ are $\mathcal{S}_2 = \{r_8, r_9, r_{10}\}$ and $\mathcal{S}_3 = \{r_{20}, r_{21}\}$ respectively.

2.1.2 Synchronized Comparative Review Sets Selection. The above formulation in Equation 1 relates the comparative items through their respective commonality in aspects with the target item. This could inadvertently results in a situation where different comparative items cover different aspects of the target item. To address this, we need to incorporate the direct commonality between any pair of comparative items.

Problem 2. Synchronized Comparative Review Sets Selection (CompaReSetS+). *We are given a collection of n items $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, where p_1 is regarded as the target item and p_2 to p_n as comparative items. Every item p_i has a collection of reviews $\mathcal{R}_i$, a target opinion vector τ_i. For a target aspect vector Γ and an integer number m, find $\mathcal{S}_i \subseteq \mathcal{R}_i$ for every item p_i such that $|\mathcal{S}_i| \leq m$ and*

$$\sum_{i=1}^{n} \Delta(\tau_i, \pi(\mathcal{S}_i)) + \lambda^2 \sum_{i=1}^{n} \Delta(\Gamma, \phi(\mathcal{S}_i)) + \mu^2 \underbrace{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Delta(\phi(\mathcal{S}_i), \phi(\mathcal{S}_j))}_{\textit{distance among items}} \quad (5)$$

is minimized.

Where $\mu \geq 0$ coefficient controls the contribution of comparisons among items towards the overall objective.

Figure 2b illustrates a CompaReSetS+ example across $\{p_1, p_2, p_3\}$. Selecting $m \leq 3$ reviews, the optimal sets of $\mathcal{R}_2$ and $\mathcal{R}_3$ are $\mathcal{S}_2 = \{r_{15}, r_{16}, r_{17}\}$ and $\mathcal{S}_2 = \{r_{20}, r_{21}\}$ respectively, both contain aspect *quality*, enabling comparison among the three items. These are different from those of CompaReSetS in Figure 2a which do not have the aspect *quality* in common.

2.2 Algorithms

Solving CompaReSetS and CompaReSetS+ optimally is intractable. Our work generalizes the Characteristic Review Selection (Crs), which has been shown to be NP-complete [15]. Crs can be seen as a special case where the number of items is 1 and $\lambda = 0$. When there are two or more items, users seek information to compare among these items for consideration. By virtue of the reducibility of our more general formulation to this special case, CompaReSetS is also NP-complete. For only one item, solving CompaReSetS+ is similar to solving CompaReSetS. Thus, CompaReSetS+ is also NP-complete.

We design heuristic algorithms using Integer-Regression algorithm as described in [15] to approximate the proposed formulations CompaReSetS and CompaReSetS+. The core idea is to find a logical vector indicating which review should be selected to approximate the value of the target vector in which the reviews are presented as a weighted matrix of aspect and opinions, which will be discussed later in this section. The Integer-Regression algorithm [15] solves Characteristic Review Selection (Crs) problem for a single item by first solving the continuous version of the optimization problem, which is a linear regression problem (see Figure 3a), then transforming the continuous solution into the closest discrete one. This well-known strategy has been shown to be effective for combinatorial optimization problems.

(CompaReSetS) Let ω be a $|\mathcal{R}_i|$-dimensional vector, each element is $\omega_j \in \{0, 1\}$, where $\omega_j = 1$ indicates the review $r_j \in \mathcal{R}_i$ being selected into the solution set $\mathcal{S}_i$. For an item p_i, let W be an $(m+z) \times |\mathcal{R}_i|$ matrix, in which each entry $W_{i'j} = 1$ iff opinion $o_{i'}$ appears in review r_j and $W_{(m+i'')j} = \lambda$ iff aspect $a_{i''}$ appears in review r_j. With the target vector be $[\tau_i; \Gamma]$, same Integer-Regression algorithm could be applied with the running time complexity of $O(m^3 + n \times m)$.

Working example 2. For a visual illustration, refer to Figure 3b. Specifically, taking $\mathcal{R}_1$ in Figure 2a as an example, the target vector is now $[\tau_1; \Gamma] = \left(\frac{2}{6}, \frac{4}{6}, \frac{2}{6}, \frac{2}{6}, \frac{2}{6}, \frac{2}{6}, 0, 0, 0, 0, \frac{6}{6}, \frac{4}{6}, \frac{4}{6}, 0, 0\right)$. Solving linear regression for $W \cdot \mathrm{x} = [\tau_1, \Gamma]$ with $\lambda = 1$, we get a general solution $\mathrm{x} = \left(\frac{1}{3} - t, \frac{1}{3} - t, \frac{1}{3} - t, \frac{1}{3} - t, t, t, t\right)$. For $m = 3$

Algorithm 1 Integer-Regression Algorithm solving CompaReSets+

Input: $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$, these sets are retrieved by solving CompaReSets on $p_1, p_2, \ldots, p_n$

```
for i = 1...n do                                                    ▷ Iterate through every item
    min_Δ = ∞
    Υ = [τ_i; Γ; φ(S_1); ...; φ(S_{i−1}); φ(S_{i+1}); ...; φ(S_n)]   ▷ Begin Integer-Regression algorithm for item p_i
    Constructing V as a (m + n × z) × |R_i| matrix, each V_{i'j} = 1 iff opinion o_{i'} appears in r_j, V_{(m+i'')j} = λ and V_{(m+t×zi'')j} = μ iff aspect a_{i''}
      appears in review r_j for t ∈ {1, 2, ..., n − 1}
    Ṽ, {c_1, ..., c_q} = DeduplicateColumns(V)                     ▷ c_i is the number of duplication columns corresponding to column i of Ṽ
    for ℓ = 1...m do
        x = NOMP(Ṽ, Υ)                                              ▷ Find x such that Δ(Ṽ, Υ) is small, 0 < ||x||_1 ≤ ℓ. Refer to [15] for NOMP.
        Find ṽ ∈ Z^q that minimizes || ṽ/||ṽ||_1 − x/||x||_1 ||_1, s.t. ∀i, ṽ_i ≤ c_i, ||ṽ_i||_1 ≤ m
        Map ṽ back to v using {c_1, ..., c_q} to form the selected reviews set S_i
        if Δ(Υ, [π(S_i); φ(S_i); φ(S_1); ...; φ(S_{i−1}); φ(S_{i+1}); ...; φ(S_n)]) < min_Δ then
            min_Δ = Δ(Υ, [π(S_i); φ(S_i); φ(S_1); ...; φ(S_{i−1}); φ(S_{i+1}); ...; φ(S_n)])
            S_i = S_i
return S_1, S_2, ..., S_n
```

we can set $t = \frac{1}{3}$ and achieve a solution $\mathrm{x} = \left(0, 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$. Transforming x into its discrete version, we get $\omega = (0, 0, 0, 0, 1, 1, 1)$ indicating an optimal set of $\mathcal{R}_1$ is $\mathcal{S}_1 = \{r_5, r_6, r_7\}$. If we set $m \geq 4$, by setting $t = 0$, we get $\mathrm{x} = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0\right)$ and we can get another optimal set $\mathcal{S}_1 = \{r_1, r_2, r_3, r_4\}$ that also has identical opinion vector and aspect vector as τ_i and Γ respectively.

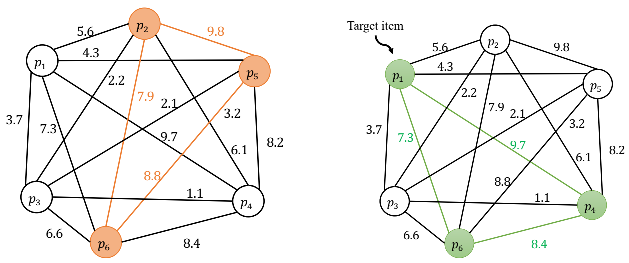
(CompaReSets+) After we achieve selected sets for each item in the collection of items $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ from CompaReSets, we can incorporate the distance among items for further optimization, that eventually solves the CompaReSets+ (Equation 5) problem simultaneously (see Algorithm 1). In particular, we alternate the following process for each item p_i, for $i = 1$ to n. Let V be a $(m + n \times z) \times |\mathcal{R}_i|$ matrix, in which each entry $V_{ij} = 1$ iff opinion o_i appears in review $r_j \in \mathcal{R}_i$, $V_{(m+i')j} = \lambda$ and $V_{(m+t \times z+i')j} = \mu$ iff aspect $a_{i'}$ appears in review r_j, for $t \in \{1, 2, \ldots, n-1\}$. And the target vector for optimization, denoted Υ, is the concatenation of τ_i, Γ, and all other aspect distribution vectors of the selected sets of reviews of other items except item p_i, i.e., $\phi(\mathcal{S}_1), \ldots, \phi(\mathcal{S}_{i-1}), \phi(\mathcal{S}_{i+1}), \ldots, \phi(\mathcal{S}_n)$. Figure 3c visualizes the construction matrix V and target vector Υ. We select a subset reviews for item p_i, finding v by minimizing $\Delta(\Upsilon, Vv)$, where v is the indicator for which review to be selected (similar to ω). For CompaReSets+, we perform this algorithm for every item $p_i \in \mathcal{P}$. Which brings the total complexity to $O((m^3 + |\bar{\mathcal{R}}| \times m) \times n)$, where $|\bar{\mathcal{R}}|$ is the average number of reviews per item.

3 CORE LIST OF COMPARATIVE ITEMS

One insight that we draw is that the objective of CompaReSets+ effectively captures the pairwise similarities between the comparative items as well as their respective similarity to the target item. Intuitively, not all comparative items are equally similar. In the event that the initial list of comparative items is long, we may need to narrow it down to a shorter list to make it easier for the end user to read their reviews. Specifically, we are interested in a list of k items that are most similar to each other including the target item.

3.1 Problem Formulation

After solving CompaReSets+ problem, the distance between two items p_i and p_j is $d_{ij} = \Delta(\tau_i, \pi(\mathcal{S}_i)) + \Delta(\tau_j, \pi(\mathcal{S}_j)) + \lambda^2 \Delta(\Gamma, \phi(\mathcal{S}_i)) + \lambda^2 \Delta(\Gamma, \phi(\mathcal{S}_j)) + \mu^2 \Delta(\phi(\mathcal{S}_i), \phi(\mathcal{S}_j))$. Intuitively, the closer the distance between two items p_i and p_j, the more similar they are.



(a) HkS solution **(b) TargetHkS solution**

Figure 4: Finding the heaviest 3-subgraph example

We can construct a complete graph G in which each item is a vertex and every pair of items p_i and p_j are connected by an edge with a weight $w_{ij} = \max_{p_{i'}, p_{j'} \in \mathcal{P}, i' \neq j'} d_{i'j'} - d_{ij}$ to turn a notion of distance into similarity. To narrow down the long list of items, we seek top-k items that are most similar to each other including the target item. This is equivalent to finding the heaviest clique in the graph G consisting k nodes including p_1 (the target item). We formally define the problem of selecting the core list of comparative items as follows:

Problem 3. Target-Oriented Heaviest k-Subgraph (TargetHkS). *Given a collection of items $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, each item p_i is corresponding to a vertex in graph G, the edges are the similarity between any two items (vertices) p_i and p_j is w_{ij}, $i \neq j$. The target item is p_1. Find a subgraph (subset of items) $\rho \subseteq \mathcal{P}$ such that $|\rho| = k$, $p_1 \subset \rho$, and*

$$\sum_{p_i, p_j \in \rho, i \neq j} w_{ij} \tag{6}$$

is maximized.

The TargetHkS problem is related yet distinct to finding Heaviest k-Subgraph (HkS) [19] because we seek to find the heaviest k-subgraph that includes the target item. When we solve TargetHkS with every vertex as the target item, we will eventually find the optimal solution for the HkS problem. Figure 4 shows an example of finding target-oriented heaviest 3-subgraph problem. The three items including target item which have the heaviest weight is $\{p_1, p_4, p_6\}$ with a weight of 25.4 is the solution for TargetHkS. Though the set $\{p_2, p_5, p_6\}$ with the heaviest weight of 26.5 is the solution for HkS problem, this solution is excluded from TargetHkS as it does not include the target item p_1.

3.2 Integer Linear Program

Here we express the optimal formulation of the TargetHkS problem via Integer Linear Programming (ILP), named $\text{TargetHkS}_{\text{ILP}}$. Let γ_i be a binary indicator whether the node p_i is a part of the solution set ρ. The weight of the edge connecting two items p_i and p_j is w_{ij}. Equation 7a is the objective to maximize the total weight (similarity) of the selected items, which is equivalent to Equation 6. Constraints at Equation 7b and Equation 7c ensure there will be exactly k items would be selected and the target item p_1 would always be a part of the solution set ρ.

$$\text{max:} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \gamma_i \gamma_j w_{ij} \tag{7a}$$

$$\text{s.t:} \sum_{i=1}^{n} \gamma_i = k \tag{7b}$$

$$\gamma_1 = 1 \tag{7c}$$

$$\gamma_i \in \{0,1\}, \forall i \in \{2,3,\ldots,n\} \tag{7d}$$

Solving $\text{TargetHkS}_{\text{ILP}}$ with $k \in \{1,2,n\}$ is trivial. However, it may be intractable for large problem sizes when $2 < k < n$.

Lemma 3.1. *The* $\text{TargetHkS}_{\text{ILP}}$ *is NP-hard.*

Proof. The proof sketch is based on the reduction from vertex cover (known to be NP-hard). Vertex cover finds the minimum set of vertices in a graph, such that all the edges in the graph are covered by at least one of the vertices in this set. We reduce vertex cover to TargetHkS where $w_{ij} = 1, \forall i \neq j$. Given that the constraints limit γ_i to either 0 or 1, any feasible solution to the TargetHkS problem is a subset of vertices. If we solve the problem for all $k \in \{1,2,\ldots,n\}$, we arrive at a solution for vertex cover with the minimum set of vertices. □

3.3 Greedy Algorithm

As the problem is NP-hard, we design a heuristic approach to solve TargetHkS problem, named $\text{TargetHkS}_{\text{Greedy}}$ (see Algorithm 2). In particular, we first include the target item in the solution set. Then we select the next item $p_{i'}$ that maximizes $\sum_{p_i,p_j \in \rho \cup \{p_{i'}\}, i \neq j} w_{i,j}$ one-by-one until k items being selected. Beside being efficient, this algorithm also proves to be effective in practice (see subsection 4.3). This algorithm iterates $k-1$ times to select the remaining items excluding the target item. The running time to compute the total weight of the current possible solution $\rho \cup \{p_{i'}\}$ is $O(|\mathcal{P}| \times |\rho|)$. Thus, the total running time is $O((k-1) \times |\mathcal{P}| \times |\rho|)$.

Algorithm 2 Greedy algorithm: $\text{TargetHkS}_{\text{Greedy}}$

Input: $\mathcal{P}, w_{ij}, k$;
1: $\rho = \{p_1\}$
2: **for** $j' = 2...k$ **do**
3: $\quad p_{i'} = \text{argmax}_{p_{i'} \in \mathcal{P}} \sum_{p_i,p_j \in \rho \cup \{p_{i'}\}, i \neq j} w_{i,j}$
4: $\quad \rho = \rho \cup \{p_{i'}\}$
5: $\quad \mathcal{P} = \mathcal{P} \setminus \{p_{i'}\}$
6: **return** $\mathcal{S}_i$

4 EXPERIMENTS

As this is a novel formulation, the main goal is to test the hypothesis that selecting reviews for multiple items jointly results in a better selection than doing so separately for comparative purposes.

Table 2: Data statistics

	Dataset		
	Cellphone	Toy	Clothing
#Product	10,429	11,924	23,033
#Reviewer	27,879	19,412	39,387
#Review	194,439	167,597	278,653
#Target Product	9,207	11,004	21,128
Avg. #Comparison Product	25.57	34.33	12.03
Avg. #Review per Product	18.64	14.06	12.10

4.1 Setup

4.1.1 Datasets. We rely on the publicly available Amazon Product Review Dataset[3] [7]. The comparison products are extracted from the product metadata in which each product contains a list of "also bought" products for comparison. While quantitative experiments alone could be run on arbitrary number of datasets, a proper user study (as we have conducted) would only be meaningful if participants could understand the reviews and make sense of the comparison. For this reason, we select three product categories of diverse nature, and yet they can be appreciated by regular users, including: Cell Phones and Accessories (Cellphone), Toys and Games (Toy), Clothing (Clothing). Table 2 summarizes basic statistics of the datasets. That the average number of comparison items could be as high as 30+ for Toys dataset, motivates why we seek to narrow down to a shorter list. We also note that every target item corresponds to an independent instance of the problem. Solving multiple target items can be done in parallel. A larger dataset that involves more items does not necessarily mean that the problem is more difficult to solve, as we apply our solution to every problem instance, not the whole dataset at once.

Sentiment data are acquired from [18] in which the authors use a frequency-based approach that follows Gao et al. [5] to extract aspects from reviews. In particular, using Microsoft Concepts[4] as aspects, the authors first retrieve top-2000 most frequently mentioned in reviews, sort them by their correlations with the ratings, and keep only top-500. Certainly, other approaches can be applied to extract aspect sentiment from product reviews, or other means of aspect (e.g. attributes, features, etc.) are also applicable. In any case, we consider these as a given.

4.1.2 Baselines. To our best knowledge, this is the first work on selecting comparative sets of reviews across multiple products. The closest baseline to the multi-product CompaReSetS and CompaReSetS+ is the single-product Characteristic-Review Selection (Crs) [15][5]. We also compare to a heuristic which greedily selects reviews one-by-one such that the selected review minimizes the overall distance cost (i.e., Equation 3), named $\text{CompaReSetS}_{\text{Greedy}}$. We also compare to Random, randomly samples review one-by-one until m reviews have been selected.

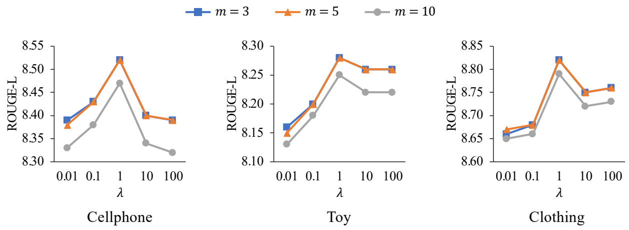
4.1.3 Metrics. We measure the alignment of the selected reviews using ROUGE [21], a well-known metric for text matching and text summarization, to assess how well the selected reviews from one item be similar to another item. Since each item may

[3] http://jmcauley.ucsd.edu/data/amazon/
[4] https://concept.research.microsoft.com/
[5] We use their best-performing algorithm as baseline. Though that was also based on a form of integer regression, it was significantly different from our method due to different objectives.

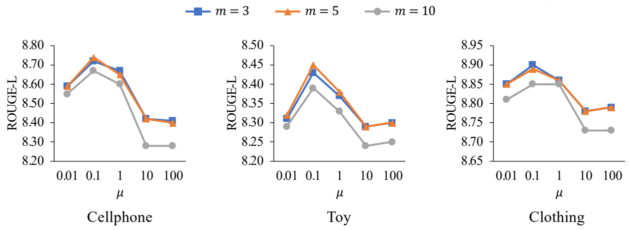
Table 3: Comparison to Baselines: Review alignment measurement for comparative review sets selection

Dataset	Algorithm	(a) Target Item vs Comparative Items									(b) Among Items								
		$m = 3$			$m = 5$			$m = 10$			$m = 3$			$m = 5$			$m = 10$		
		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Cellphone	Random	15.03	1.12	7.92	15.02	1.12	7.92	15.02	1.12	7.92	14.74	1.05	7.81	14.73	1.05	7.82	14.73	1.05	7.81
	Crs	15.99	1.28	8.44	15.99	1.28	8.45	15.92	1.27	8.41	15.79	1.23	8.40	15.79	1.23	8.40	15.69	1.20	8.36
	CompaReSetS$_{Greedy}$	15.07	1.12	8.05	15.08	1.12	8.03	15.06	1.12	7.98	15.07	1.12	8.05	15.08	1.12	8.03	15.06	1.12	7.98
	CompaReSetS	16.28	1.36	8.52	15.99	1.28	8.45	16.22	1.35	8.47	15.86	1.23	8.50	15.86	1.23	8.49	15.79	1.22	8.44
	CompaReSetS+	**16.31***	**1.37***	**8.72***	**16.48***	**1.40***	**8.74***	**16.43***	**1.38***	**8.67***	**15.93***	**1.24***	**8.75***	**16.09***	**1.28***	**8.73***	**16.04***	**1.26***	**8.65***
Toy	Random	15.86	1.33	7.98	15.82	1.32	7.97	15.84	1.32	7.98	15.55	1.25	7.90	15.54	1.25	7.90	15.54	1.25	7.89
	Crs	16.26	1.45	8.10	16.26	1.45	8.12	16.28	1.45	8.13	15.97	1.37	8.04	15.98	1.36	8.07	15.99	1.37	8.07
	CompaReSetS$_{Greedy}$	15.92	1.34	8.13	15.89	1.33	8.06	15.87	1.33	8.01	15.92	1.34	8.13	15.89	1.33	8.06	15.87	1.33	8.01
	CompaReSetS	16.58	1.52	8.28	16.59	1.52	8.28	16.57	1.52	8.25	16.07	1.37	8.25	16.07	1.37	8.25	16.06	1.37	8.22
	CompaReSetS+	**16.67***	**1.54***	**8.43***	**16.72***	**1.55***	**8.45***	**16.71***	**1.55***	**8.39***	**16.21***	**1.39***	**8.40***	**16.26***	**1.40***	**8.41***	**16.24***	**1.40***	**8.35***
Clothing	Random	15.56	1.17	8.46	15.53	1.17	8.45	15.54	1.17	8.46	15.32	1.13	8.37	15.32	1.13	8.38	15.32	1.13	8.38
	Crs	16.37	1.31	8.83	16.37	1.31	8.83	16.33	1.32	8.80	16.18	**1.26**	8.78	16.17	**1.27**	8.77	16.13	**1.27**	8.74
	CompaReSetS$_{Greedy}$	15.56	1.17	8.52	15.59	1.17	8.52	15.57	1.17	8.48	15.56	1.17	8.52	15.59	1.17	8.52	15.57	1.17	8.48
	CompaReSetS	16.59	**1.36***	8.82	16.58	**1.36***	8.82	16.55	**1.36***	8.79	16.12	1.25	8.74	16.12	1.26	8.74	16.10	1.25	8.71
	CompaReSetS+	**16.67***	**1.36***	**8.90***	**16.66***	**1.36***	**8.89***	**16.62***	**1.36***	**8.85***	**16.20***	1.25	**8.82***	**16.20***	1.25	**8.81***	**16.17***	1.26	**8.77***

*denotes statistically significant improvements over the second best approach (p-value < 0.05). Highest values are in **bold**.

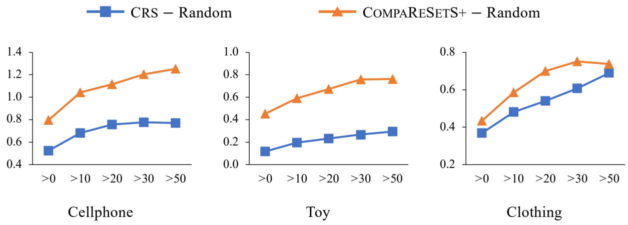


(a) ROUGE-L measure of CompaReSetS with varying λ

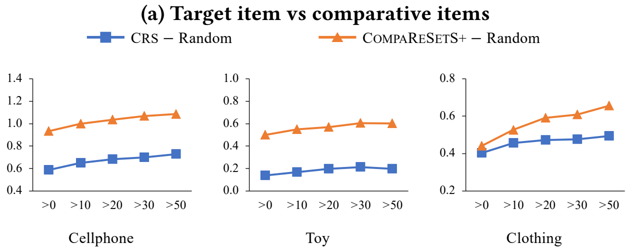


(b) ROUGE-L measure of CompaReSetS+ with varying μ

Figure 5: Review alignment measurement between target item vs comparative items



(a) Target item vs comparative items



(b) Among items

Figure 6: ROUGE-L measure of selected sets of reviews of CompaReSetS+ vs Random and Crs vs Random

have multiple reviews in the selected sets, we measure the similarity between each pair of reviews (two reviews coming from different items) and report the average score. Taken into account word-level as well as phrase-level, we report F1-score of ROUGE-1 or R-1 (unigrams), ROUGE-2 or R-2 (bigrams), and ROUGE-L or R-L (longest common subsequence). ROUGE metrics range between 0 and 1, with higher scores indicating higher similarity between one review and another review.

4.1.4 Detail Settings. With the availability of data, we investigate our proposed CompaReSetS and CompaReSetS+ with a setting where the target aspect distribution vector $\Gamma = \phi(\mathcal{R}_1)$ reflects the target item aspect distribution and the target opinion vector $\tau_i = \pi(\mathcal{R}_i)$ reflects the opinion distribution of item p_i. The maximum number of reviews to be selected are $m \in \{3, 5, 10\}$. We tune λ in a candidate set of $\{0.01, 0.1, 1, 10, 100\}$ for CompaReSetS, which achieves the best performance on ROUGE-L score with $\lambda = 1$ (see Figure 5a). Thus, we set $\lambda = 1$ and tune the coefficient μ of CompaReSetS+ in a candidate set of $\{0.01, 0.1, 1, 10, 100\}$. CompaReSetS+ achieves the best performance on ROUGE-L score with $\mu = 0.1$ (see Figure 5b). The results are consistent across datasets. After review sets selection process, we perform selecting core list of comparative items problem by solving TargetHkS problem. We set $k = m$ for simplicity, the number items k is the same as the maximum number of reviews m.

4.2 Comparative Review Sets Selection

4.2.1 Review Alignment Between Target Item and Comparative Items. Here we assess how well the selected sets of reviews of the comparison items align to those of the target item. As reported in Table 3a, CompaReSetS+ algorithm achieve best performance in all ROUGE measures across all datasets. Moreover, CompaReSetS does enhance the alignment of the selected review sets between the target item and comparison items, which also the second best performing algorithm. CompaReSetS achieves lower ROUGE

Table 4: Review alignment (ROUGE-L) between target item and comparative items across opinion definitions

Algorithm	Opinion definition		
	binary (default)	3-polarity	unary-scale
CRS	8.44	8.43	7.59
COMPARESETS$_{\text{Greedy}}$	8.05	8.05	8.16
COMPARESETS	8.52	8.52	8.52
COMPARESETS+	8.72	8.72	8.25

scores than COMPARESETS+ in most cases, highlighting that selecting comparative sets of reviews synchronously is better for comparison than doing so separately. The reviews selected by COMPARESETS$_{\text{Greedy}}$ are more aligned than those of the Random baseline. Nevertheless, this approach achieves lower review alignment measurement than CRS, showing the advantage of using Integer-Regression algorithm over a simple greedy algorithm in review selection problem.

We intuit that products may vary in terms of problem difficulty. If a product has few reviews, different selection methods may not yield significantly different results. On the other hand, if a product has many reviews, the combinatorial explosion yields numerous subsets of reviews, making the task more challenging and introducing greater variability among methods. To test this, in Figure 6a, we plot the performance gap between COMPARESETS+ and Random (the orange lines) for items with different number of reviews in which the performance gap is the different of ROUGE-L measure of the solution sets generated by COMPARESETS+ and Random. Evidently, the higher the number of reviews, the larger the performance gap. Similar trend is observed for CRS (blue line), with lower performance than our proposed COMPARESETS+.

4.2.2 Review Alignment Among Comparative Items. Our proposed COMPARESETS+ problem focuses on selecting comparative sets of reviews. Ideally, the selected sets of reviews, when comparing one item to other items, are "well-aligned" for better comparisons. Table 3b shows that the proposed COMPARESETS+ consistently outperforms the CRS baseline significantly across all the datasets in terms of ROUGE-L. This suggests that our proposed algorithm works as expected. We also observe a similar trend as mentioned earlier when we compare to only the target product, the performance gaps between COMPARESETS+ and Random when measuring the review alignment among products are also larger for products with higher number of reviews when measuring against Random (see Figure 6b).

4.2.3 Generalization Beyond Positive and Negative Opinions. Although we formulate the opinion vector using a *binary* scale of sentiments in section 2 (similar to [15]), the proposed method can be applied with a broader range sentiments in general. In addition to *binary* opinions, here we further discuss two other definitions:

3-polarity: besides *positive* and *negative* sentiment, there will be an additional *neutral* sentiment, $\pi(\mathcal{S}_i) \in \mathbb{R}_+^{3z}$.

Unary-scale: each aspect will be associated with a score that reflect the sentiment polarity, $\pi(\mathcal{S}_i) \in \mathbb{R}_+^{z}$. We use a Sigmoid function $\frac{1}{1+e^{-s}}$ to transform the aggregated sentiment (i.e., sum) into $[0, 1]$ scale, where s is the total sentiment of the given aspect.

Table 4 shows that the proposed COMPARESETS and COMPARESETS+ perform best overall. ROUGE-L measures of CRS are higher for *binary* and *3-polarity*. For *unary-scale*, CRS underperforms the Random baseline (ROUGE-L is 7.92) while COMPARESETS and

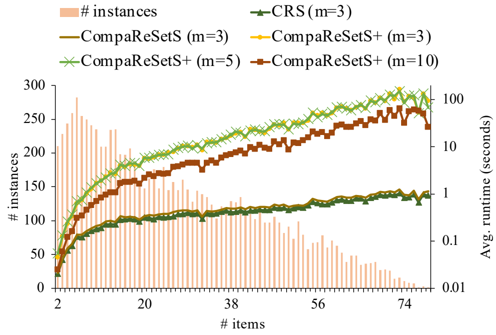


Figure 7: Average runtime with different number of comparative items on Cellphone data

Table 5: Performance ratios over TARGETHKS$_{\text{ILP}}$ (%)

Dataset	k	#Optimal Solution	Objective Value Ratio	
			TARGETHKS$_{\text{Greedy}}$	Random
Cellphone	3	100.00	-0.00005	-21.97
	5	99.59	-0.00008	-22.14
	10	80.85	-0.00002	-18.96
Toy	3	100.00	-0.00009	-19.39
	5	99.00	-0.00015	-21.04
	10	66.78	0.00147	-20.14
Clothing	3	100.00	-0.00013	-24.59
	5	99.98	-0.00013	-23.28
	10	98.45	-0.00004	-18.00

COMPARESETS+ are better generally. In the consequence experiments, we will only focus on *positive* and *negative* opinions.

Besides the above mentioned definition of opinion vector, we can also use other alternatives, such as learned aspect-level preference vectors from another model (e.g., such as EFM [42] or MTER [34]) of a reviewer on a given item, where multiple reviews can be aggregated (e.g., average, max, MLP, etc.). Without loss of generality, we leave this for future exploration.

4.2.4 Runtime with Different Number of Items. Figure 7 illustrates the runtime (seconds) of different algorithms on Cellphone data. Similar to CRS, COMPARESETS is quite efficient with integer regression algorithm, which solves every problem instances in around 1 second. COMPARESETS+ is expected to be slower as it repeatedly runs integer regression algorithm for every items (time complexity is linear to the number of items). For COMPARESETS+, running time when $m = 5$ is roughly similar to that of $m = 3$. When $m = 10$, the running time is faster, indicating that increasing the number of reviews to be selected does not necessary make the problem more difficult to be solved. Similar trends are observed across datasets.

4.3 Core List of Comparative Items

As a secondary objective, after retrieving the sets of reviews from COMPARESETS+, we further narrow down the list of comparative products to ease the cognitive load on consumers. This section assesses the effectiveness of the TARGETHKS approximation.

4.3.1 Optimal vs Approximation. For the optimal TARGETHKS$_{\text{ILP}}$, when limiting the running time to be 60 seconds, the Gurobi[6] solver could still solve optimally for virtually all problem instances when $k \in \{3, 5\}$. For the larger $k = 10$, the percentages vary from two-thirds to close to full across datasets (see Table 5),

[6]https://www.gurobi.com/

Table 6: Review alignment measurement for core list of comparative items

Dataset	Algorithm	(a) Target Item vs Comparative Items									(b) Among Items								
		$k=m=3$			$k=m=5$			$k=m=10$			$k=m=3$			$k=m=5$			$k=m=10$		
		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Cellphone	Random	16.34	1.39	8.74	16.49	1.39	8.74	16.42	1.39	8.67	16.20	1.33	8.75	16.20	1.30	8.72	16.06	1.27	8.65
	Top-k similarity	16.71	1.50	8.94	16.76	1.48	8.91	16.61	1.43	8.78	16.71	1.46	9.09	16.58	1.41	9.00	16.34	1.34	**8.84**
	TargetHkS$_{\text{Greedy}}$	16.91	1.54	8.91	16.88	**1.50**	**8.93**	**16.65**	**1.44**	**8.79**	16.95	1.53	9.02	16.77	**1.45**	**9.03**	**16.38**	**1.35**	**8.84**
	TargetHkS$_{\text{ILP}}$	**16.93**	**1.54**	**8.93**	**16.89**	**1.50**	**8.93**	16.64	**1.44**	8.78	**16.96**	**1.53**	**9.04**	**16.78**	**1.45**	**9.03**	**16.38**	**1.35**	**8.84**
Toy	Random	16.72	1.56	8.46	16.72	1.56	8.46	16.70	1.54	8.38	16.53	1.49	8.44	16.41	1.45	8.44	16.28	1.41	8.35
	Top-k similarity	16.93	1.64	8.63	16.90	1.60	8.59	16.83	1.57	8.47	16.87	1.61	8.73	16.75	1.54	8.73	16.55	1.46	8.57
	TargetHkS$_{\text{Greedy}}$	17.11	1.67	8.67	16.96	1.62	8.61	16.84	1.57	**8.48**	17.12	1.67	8.78	16.91	1.60	8.76	16.61	1.48	8.59
	TargetHkS$_{\text{ILP}}$	**17.14**	**1.71**	**8.70**	**17.00**	**1.64**	**8.63**	**16.85**	**1.58**	**8.48**	**17.16**	**1.70**	**8.80**	**16.95**	**1.61**	**8.79**	**16.62**	**1.49**	**8.60**
Clothing	Random	16.67	1.37	8.90	16.69	1.37	8.90	16.62	1.36	8.84	16.46	1.31	8.87	16.32	1.28	8.84	16.19	1.26	8.77
	Top-k similarity	16.81	1.41	8.99	16.74	1.38	8.95	16.65	1.36	**8.87**	16.64	1.36	9.00	16.43	1.30	8.95	16.24	**1.27**	8.84
	TargetHkS$_{\text{Greedy}}$	16.91	1.41	9.02	16.78	1.38	8.95	**16.66**	**1.37**	**8.87**	16.79	**1.38**	9.04	16.53	**1.32**	8.98	**16.27**	**1.27**	**8.85**
	TargetHkS$_{\text{ILP}}$	**16.94**	**1.42**	**9.03**	**16.80**	**1.39**	**8.96**	**16.66**	**1.37**	**8.87**	**16.82**	**1.38**	**9.06**	**16.55**	**1.32**	**8.99**	**16.27**	**1.27**	**8.85**

Highest values are in **bold**.

which is in line to the average number of comparison products reported in Table 2. In other words, the higher number of comparison products, the more difficult problem is to solve. We are then interested in seeing how well the approximation could approach the optimal. Table 5 also reports the ratios of the difference between the objective values achieved by approximation algorithms ($\Omega_{\text{approximation}}$) and the optimal TargetHkS$_{\text{ILP}}$ ($\Omega_{\text{TargetHkS}_{\text{ILP}}}$),

$$\text{Objective Value Ratio} = \frac{\Omega_{\text{approximation}} - \Omega_{\text{TargetHkS}_{\text{ILP}}}}{\Omega_{\text{TargetHkS}_{\text{ILP}}}} \quad (8)$$

Where $\Omega_{\text{algorithm}} = \sum_{\rho} \sum_{p_i, p_j \in \rho, i<j} w_{ij}$ is the total weight of all edges (total similarity) across all solutions, which ρ is the solution of a given algorithm (i.e., TargetHkS$_{\text{ILP}}$, TargetHkS$_{\text{Greedy}}$, or Random) on a problem instance. Overall, the heuristic approach TargetHkS$_{\text{Greedy}}$ objective values are almost similar to those of TargetHkS$_{\text{ILP}}$. Comparing that to Random, selecting $k-1$ products randomly as the target product p_1 is always belong to the solution set, the Random approach performs poorly with a big gap, with objective value reduction ratio $>$ 20% overall. TargetHkS$_{\text{Greedy}}$ can sometimes be better than TargetHkS$_{\text{ILP}}$ (solving by Gurobi solver within 60 seconds) in the more difficult problem instances (e.g, when $k = 10$ on Toy data).

4.3.2 Review Alignment. To further assess the effectiveness of the proposed TargetHkS approximation, we re-evaluate the similarity among selected items using ROUGE measure on their selected reviews. For parity, the same sets of selected reviews are from CompaReSetS+ algorithm. Table 6a shows the similarity of reviews between the target item and the comparative items. Table 6b illustrates the similarity among all items. Overall, the reviews of items selected by both TargetHkS$_{\text{ILP}}$ and TargetHkS$_{\text{Greedy}}$ are more similar than those of Random approach. And TargetHkS$_{\text{Greedy}}$'s performance approaches those of TargetHkS$_{\text{ILP}}$. We further add Top-k similarity baseline, selecting top-k highest similar items to the target item. Our proposed TargetHkS still performs best in most cases. Except with high value $k = 10$, Top-k similarity's performance approaches those of TargetHkS.

4.4 Case Studies

To gain a better intuition on the proposed approach, we present several illustrative case studies on several product categories.

Figure 8: Selected sets of reviews of a Cellphone instance

Figure 8 shows a case study of car chargers. These three products are the top-3 most similar items selected by TargetHkS$_{\text{ILP}}$ from a list of total 10 also bought products. Each and every product has 3 reviews. All of the reviews of the three products discuss about a common aspect *charger*. The selected set of reviews for each product cover diverse sentiments on various aspects that also appear in the other two products. One aspect they touch on is the use of the charger for iPhone (mentioned in the selected reviews for all products). Another aspect is its in-car use (also universally mentioned). The second review of the first product and the third review of the third product discuss durability. The third review of the first product and the first review of the third product discuss how quickly it charges. This shows how synchronizing the review selection across comparative products help to feature reviews that allow better comparisons.

Figure 9: Selected sets of reviews of a Toy instance

Figure 9 shows a case study of three 1000-Piece Puzzle products of Ravensburger Puzzles brand. These top-3 most similar items are selected from a list of total 18 products by TargetHkS$_{\text{ILP}}$. The selected sets of reviews cover numerous aspects that are relevant for comparison. One of them is the puzzle itself (discussed in every review). The second review of the first product, all the reviews of the second product, and all the reviews of the third product discuss quality. While the first product has an additional augmented reality feature and the second product can glow in the dark, the third product is just another conventional puzzle. This further shows how synchronizing the review selection across comparative products help to feature reviews that allow better comparison.

Figure 10 shows a case study of Sandal products. These products are top-3 most similar items selected from a list of 7 products by TargetHkS$_{\text{ILP}}$. The selected sets of reviews discussed various aspects that are related. Size, fit, and color are being discussed in all items. While the first and the third product are fit true to size, the second sizing is a bit smaller. The second review of the first product and the first review of the second product emphasize that these shoes are comfortable to walk on all day. This further shows how synchronizing the review selection across comparative products help to feature reviews that allow better comparison.

Figure 10: Selected sets of reviews of a Clothing instance

4.5 User Study

In this section, we conduct a qualitative study on the efficacy of the selected sets of reviews from human perspective. First, we select 3 different products from each category to get 9 examples in total. Each example consists of one target product and two other products which are most relevant selected by TargetHkS$_{\text{ILP}}$ on reviews selected by CompaReSetS+. Then we design 3 independent surveys, each containing 9 examples of different review selection algorithms (including CompaReSetS+, Crs, and Random algorithms) and presented blindly in random ordered (the participants do not know which sets of reviews selected by which algorithm). For parity, we only present examples which have exactly 3 selected reviews from CompaReSetS+, Crs, and Random algorithms. Involving 15 participants who are not the authors, each example is assessed by 5 participants. For each example, we ask the following three questions:

- **Q1:** How similar are the reviews among products (i.e., discussing same aspects)?
- **Q2:** Do reviews help you know more about the recommended products?
- **Q3:** Do reviews help you in comparison among products?

The three questions serve different purposes. The first question looks into similar aspects among reviews selected from different algorithms. The second looks into the appropriate of the selected reviews with the product. The third question looks into the comparative information given by the selected sets of reviews for comparing the products. Each participant answers each question by choosing from five-point Likert scale, from 1 (strongly disagree/strongly dissimilar) to 5 (strongly agree/strongly similar).

Table 7: Result analysis of user study

Algorithm	Q1	Q2	Q3	Krippendorff's $\alpha^{\dagger}$
Random	3.47	3.78	3.38	−0.039
CRS	3.69	4.07	3.64	0.050
COMPARESETS+	**3.73**	**4.18**	**3.71**	**0.299**

†Higher value means higher agreement. Highest values are in **bold**.

The overall result of the surveys is reported in Table 7. The evaluation scores of COMPARESETS+ are consistently the highest among comparison algorithms, indicating that our proposed COMPARESETS+ selects higher quality sets of reviews (reviews that are more similar among products, more informative, and better for comparison purpose). Given the abundance of reviews for each product, the average scores are > 3, which indicate the reviews are providing useful information for the appropriate products. The shortlist of products are quite similar as they are produced by the same TARGETHKS$_{\text{ILP}}$ algorithm. This explains the small gap of question 1 scores between CRS and COMPARESETS+ as the products may have many of aspects in common so that CRS easily achieves high evaluation but still lower than COMPARESETS+. The higher scores on question 2 may indicate that most reviews from all algorithms are informative to provide user with additional information to know more about the recommended product. For question 3, there is an improvement in comparative information in reviews selected by COMPARESETS+. Being aware that the number of samples of this user study is small and is insufficient for performing statistical test, we measure the Krippendorff's Alpha-Reliability coefficient [14] to assess the agreement among annotators. The reviews selected by COMPARESETS+ achieve higher $\alpha = 0.299$, indicating some level of reliability, compared to that of CRS, α is 0.05. Random approach has the lowest $\alpha = -0.039$, which is negative, showing some disagreements among annotators.

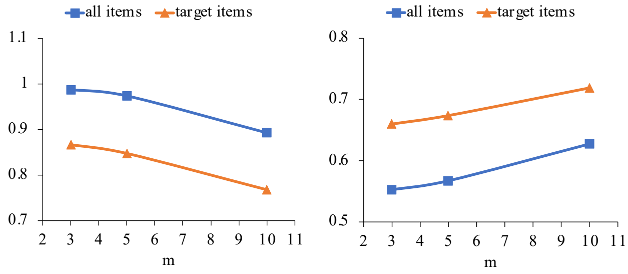
4.6 Discussion

4.6.1 Information Loss When Selecting Subsets of Reviews. Presenting a subset of reviews to the end user helps reduce information overload (less redundant and more focus). However, it may also omit some aspects that have never been discussed in the selected set of reviews due to the large number of reviews, which potentially beyond thousands, and this phenomenon is inevitable.

Figure 11 shows a clear trend that the amount of information loss reduces when increasing the number of maximum reviews to be selected m. Figure 11a illustrates the amount of information loss of the selected set of reviews for an item i via $\Delta(\tau_i, \pi(\mathcal{S}_i))$. In ideal cases, the values approach 0, meaning the selected set represents the whole set of reviews without loss of information. Figure 11b displays cosine similarity between τ_i and $\pi(\mathcal{S}_i)$,

$$\cos(\tau_i, \pi(\mathcal{S}_i)) = \frac{\tau_i \cdot \pi(\mathcal{S}_i)}{||\tau_i|| \cdot ||\pi(\mathcal{S}_i)||} \tag{9}$$

when values approaching 1 mean τ_i and $\pi(\mathcal{S}_i)$ are very similar. Another note is that the amount of information loss is larger when evaluating all items (including target item and comparative items), because the selected reviews of comparative items are skewed towards the target item for comparison with the target item, which is expected. This observation is consistent across datasets. This trend shows an inherent trade off. Increasing m reduces the amount of information loss, but it can overwhelm



(a) $\Delta(\tau_i, \mathcal{S}_i)$ **(b)** $\cos(\tau_i, \pi(\mathcal{S}_i))$

Figure 11: Information loss measurement for COMPARESETS+ on Cellphone data

the end user with too many reviews to read. Although this can be further addressed using text summarization methods, we leave it for future exploration as it is beyond the scope of this work.

4.6.2 Relation to Large Language Models (LLMs). We can now easily interact with a pretrained LLM (e.g., ChatGPT[7]). One can ask LLM to compare two piece of texts whether they are comparable (similar to entity matching). However, if we tackle our comparative sets of reviews selection problem by this approach, it may explode by the number of choices given the large number of reviews and items. Taking Cellphone data as an example, on average we have more than 18 comparative items and each item has more than 25 reviews, the average number of comparisons that we need to perform is larger than 25^{18}. The number of comparisons is even larger when we combine a subset of reviews for each item, e.g., there are $\binom{25}{3} = 2{,}300$ ways to select a subset of 3 reviews from a set of 25 reviews that requires $2{,}300^{18}$ comparisons for 18 items.

LLMs are very strong in producing generative text and are known to hallucinate. To illustrate this phenomenon, we ask ChatGPT, a chat bot from OpenAI that utilizes Generative Pretrained Transformer (i.e., GPT-3.5), in selecting comparative items and reviews for comparison purposes, given the prompt below:

```
Select three comparable items, including the
target item "<Target item's title>" as the
first item. Provide 3 reviews for each item for
comparison.
```

Figure 12 illustrates an example[8] of the given prompt command with "Paris, 1000-pieces augmented reality puzzle" item. Although they are real items and can be easily found on Google search and ChatGPT generate reviews that reads naturally, we can spot out the misinformation in the reviews, e.g., the third review of the third item "London landmarks, 1000-piece jigsaw puzzle" does not have augmented reality feature. We can also see that most of the reviews complement the products and discuss general features. In addition, it is difficult to check the authenticity of the reviews[9]. Moreover, it good to note that LLMs may return different results depending on given inputs. This motivates the search for better prompts to enable LLMs to respond more accurately, which poses another research direction to explore in the future.

[7]https://chat.openai.com/
[8]More examples can be found at https://chat.openai.com/share/4f729769-7855-4442-be70-cbc41d711116
[9]We searched on Google and found NONE of the generated reviews from real users

Prompt: Select three comparable items, including the target item "Paris, 1000-pieces augmented reality puzzle" as the first item. Provide 3 reviews for each item for comparison.

ChatGPT: Sure, here are three comparable items along with three reviews for each:

1.Paris, 1000-pieces Augmented Reality Puzzle:

1. Review 1: "This puzzle was an absolute delight to work on! The combination of traditional puzzle-solving with augmented reality made the experience so immersive. Seeing Paris landmarks come to life through my phone was truly magical."
2. Review 2: "As a puzzle enthusiast, I've tried many puzzles, but this one stands out. The quality of the pieces is excellent, and the augmented reality feature adds a whole new dimension to the puzzle-solving experience. Highly recommended for anyone who loves puzzles and technology!"
3. Review 3: "I bought this puzzle as a gift for my friend who loves Paris, and she absolutely adored it. The augmented reality aspect was a unique touch that added excitement and intrigue to the puzzle-solving process. It's a must-have for puzzle lovers and Paris enthusiasts alike."

2.New York City, 1000-piece Jigsaw Puzzle:

1. Review 1: "This New York City puzzle was challenging yet rewarding to complete. The pieces fit together perfectly, and the image of the iconic skyline is stunning. It provided hours of entertainment for the whole family."
2. Review 2: "I'm a big fan of cityscape puzzles, and this one did not disappoint. The quality of the puzzle pieces is top-notch, and the vibrant colors really bring the New York City skyline to life. Highly recommend it for puzzle enthusiasts!"
3. Review 3: "I bought this puzzle as a souvenir from my trip to New York, and it was a great way to reminisce about my time there. The puzzle pieces are sturdy, and the finished product looks fantastic. I'll definitely be framing it!"

3.London Landmarks, 1000-piece Jigsaw Puzzle:

1. Review 1: "I thoroughly enjoyed working on this London landmarks puzzle. The image captures the essence of the city beautifully, and the puzzle pieces are of excellent quality. It was both challenging and fun to complete."
2. Review 2: "This puzzle featuring London landmarks is a must-have for any puzzle enthusiast or fan of the city. The level of detail is impressive, and the pieces fit together seamlessly. I especially loved seeing iconic landmarks like Big Ben and the Tower Bridge come together."
3. Review 3: "I received this puzzle as a gift, and it quickly became one of my favorites. The image of London landmarks is captivating, and the puzzle pieces are sturdy and well-made. The augmented reality feature adds an extra layer of excitement to the puzzle-solving experience."

Figure 12: Selected items and sets of reviews by ChatGPT

5 RELATED WORK

5.1 Review Mining

Previous works have primarily focused on selecting a subset of reviews from a large collection of a single product, distinguished by their respective objectives. Lappas and Gunopulos [16] proposed to select a subset of *comprehensive* reviews that represent all aspects. Tsaparas et al. [29] expands this to cover both positive and negative opinions, to select a subset of reviews that collectively provide both the negative and positive opinions on each aspect. Lappas et al. [15] optimizes for opinion distribution, providing a subset of *characteristic* reviews that statistically capture the overall proportion of opinions of an item. In addition, Chen et al. [2], Xu et al. [36], Yu et al. [39] concern the notion of review quality. Tu et al. [31] selects personalized set of reviews which includes reviews related to the aspects important to the user.

Aside from review selection, other researchers focus on developing *review summarization* methods [8, 12, 27, 28, 40, 46], which aim to create a summary of a review corpus that is both compact and representative of the opinions it contains. However, this approach often omits the structure of reviews written by real users. Others attempt to address the problem by *review ranking* methods [6, 11, 22–24, 30, 44], which score every review to reflect their quality. However, this approach overlooks the complementarity among different reviews. In turn, helpful review prediction. Huang et al. [9] selects helpful reviews that meet user's aspect sentiment. Besides, Zhu et al. [45] selects informative tokens to produce tips from customer reviews. These lines of research are orthogonal from our proposed formulation in terms of selecting a subsets of reviews. Considering them as baselines is beyond the scope of this work and we leave this for future exploration.

5.2 Comparison Mining

Our work is also related to comparison mining. Works related to comparisons include deciding whether one product is better than another [20, 43] or identifying substitute vs. complementary products [26, 41]. McAuley et al. [26] relies on reviews and networks of co-browsed and co-purchased products. Zhang et al. [41] mines product relationships using a neural network based framework that integrates the textual content and non-textual information of online reviews. Another line of research is competitor mining [10, 17, 38], which seeks those products that are most comparable to a product. Our work is complementary to these directions. Rather than focusing on the selection of which products to compare to, we assume they are given and focus on finding the comparative sets of reviews. Furthermore, we find others produce comparative recommendation explanations via template [18] or generated text [3, 33, 37]. In terms of opinion comparison, Matakos et al. [25] analyzes a social network graph to find two different subgroups who hold different opinions.

5.3 Heaviest k-Subgraph

Our formulation in narrowing down the list of comparative items (TargetHkS problem described in section 3) is related to finding the Heaviest k-Subgraph (HkS) [1, 4, 19, 32] or maximum edge subgraph problem. In particular, Asahiro et al. [1] greedily removes a vertex with the minimum weighted-degree in the currently remaining graph, until exactly k vertices are left. Feige and Langberg [4] uses semidefinite programming. Vassilevska et al. [32] also works with node-weighted case. Letsios et al. [19] derives exact solution for HkS problem using branch and bound algorithm and applies in social media context. HkS is distinct from our proposed TargetHkS problem because TargetHkS seeks the heaviest k-subgraph that must include a given target item. Finding solution of TargetHkS for every item as target item will eventually solve HkS problem.

5.4 Recommender Systems

The field of recommender systems is orthogonal to ours in that they mainly focus on the aspect of personalization, and the key relation being considered is that between users and items, particularly in terms of preference [13]. In contrast, the key relation being considered in our work is that between items, particularly in terms of their comparability. However, there may be some basis for "recommending" items based on content similarity. Using the target item as anchoring in ranking the candidate set of items, we could select the top-k items with the highest edge weights that are connected with this target item. Another line is session-based recommendation [35], including next-item or next-basket recommendations. Next-item recommender systems receive inputs as a sequence of previous interactions. Next-basket recommender systems take past ordered/purchased history as inputs.

6 CONCLUSION

In this work, we address a novel problem of selecting comparative sets of reviews for a set of comparable items, which include one target item and other comparison items. The review selection process can be performed individually (CompaReSetS) or synchronously (CompaReSetS+). As the problem is intractable in nature, we propose an approximation based on integer regression. Beyond the review selection process, to help the users in focusing on the most comparable subset of items, we further narrow down the list of comparable items to top-k most similar items including the target item. Extensive experiments validate the efficacies of our proposed heuristic algorithms, which are further supported by a user study.

ACKNOWLEDGMENTS

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-020). Hady W. Lauw gratefully acknowledges the support by the Lee Kong Chian Fellowship awarded by Singapore Management University.

REFERENCES

[1] Yuichi Asahiro, Kazuo Iwama, Hisao Tamaki, and Takeshi Tokuyama. 2000. Greedily Finding a Dense Subgraph. *Journal of Algorithms* 34, 2 (2000), 203–221.

[2] Jiawei Chen, Hongyan Liu, Yinghui (Catherine) Yang, and Jun He. 2019. Effective Selection of a Compact and High-Quality Review Set with Information Preservation. *ACM Trans. Manage. Inf. Syst.* 10, 4 (dec 2019). https://doi.org/10.1145/3369395

[3] Jessica Echterhoff, An Yan, and Julian McAuley. 2023. Comparing Apples to Apples: Generating Aspect-Aware Comparative Sentences from User Review. *arXiv preprint arXiv:2307.03691* (2023).

[4] Uriel Feige and Michael Langberg. 2001. Approximation Algorithms for Maximization Problems Arising in Graph Partitioning. *Journal of Algorithms* 41, 2 (2001), 174–211.

[5] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. 2019. Explainable Recommendation through Attentive Multi-View Learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 3622–3629.

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

[7] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *WWW (WWW '16)*.

[8] Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*. Association for Computing Machinery, New York, NY, USA, 168–177. https://doi.org/10.1145/1014052.1014073

[9] Chunli Huang, Wenjun Jiang, Jie Wu, and Guojun Wang. 2020. Personalized Review Recommendation Based on Users' Aspect Sentiment. *ACM Trans. Internet Technol.* 20, 4, Article 42 (oct 2020), 26 pages. https://doi.org/10.1145/3414841

[10] Myungha Jang, Jin-woo Park, and Seung-won Hwang. 2012. Predictive Mining of Comparable Entities from the Web. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI'12)*. AAAI Press, 66–72.

[11] Soo-Min Kim, Patrick Pantel, Tim Chklovski, and Marco Pennacchiotti. 2006. Automatically Assessing Review Helpfulness. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP '06)*. Association for Computational Linguistics, USA, 423–430.

[12] Nagsen Komwad, Paras Tiwari, Banoth Praveen, and C. Ravindranath Chowdary. 2022. A Survey on Review Summarization and Sentiment Classification. *Knowl. Inf. Syst.* 64, 9 (sep 2022), 2289–2327. https://doi.org/10.1007/s10115-022-01728-y

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

[14] Klaus Krippendorff. 2011. Computing Krippendorff's alpha-reliability. (2011).

[15] Theodoros Lappas, Mark Crovella, and Evimaria Terzi. 2012. Selecting a Characteristic Set of Reviews. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*. Association for Computing Machinery, New York, NY, USA, 832–840.

[16] Theodoros Lappas and Dimitrios Gunopulos. 2010. Efficient Confident Search in Large Review Corpora. In *Machine Learning and Knowledge Discovery in Databases*, José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 195–210.

[17] Theodoros Lappas, George Valkanas, and Dimitrios Gunopulos. 2012. Efficient and Domain-Invariant Competitor Mining. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*. Association for Computing Machinery, New York, NY, USA, 408–416. https://doi.org/10.1145/2339530.2339599

[18] Trung-Hoang Le and Hady W. Lauw. 2021. Explainable Recommendation with Comparative Constraints on Product Aspects. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM '21)*. Association for Computing Machinery, New York, NY, USA, 967–975.

[19] Matthaios Letsios, Oana Denisa Balalau, Maximilien Danisch, Emmanuel Orsini, and Mauro Sozio. 2016. Finding heaviest k-subgraphs and events in social media. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE, 113–120.

[20] Si Li, Zheng-Jun Zha, Zhaoyan Ming, Meng Wang, Tat-Seng Chua, Jun Guo, and Weiran Xu. 2011. Product Comparison Using Comparative Relations. In *SIGIR (SIGIR'11)*. ACM.

[21] Chin-Yew Lin and Eduard Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1 (NAACL '03)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 71–78.

[22] Jingjing Liu, Yunbo Cao, Chin-Yew Lin, Yalou Huang, and Ming Zhou. 2007. Low-quality product review detection in opinion summarization. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*. 334–342.

[23] Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. 2008. Modeling and predicting the helpfulness of online reviews. In *2008 Eighth IEEE international conference on data mining*. IEEE, 443–452.

[24] Yue Lu, Panayiotis Tsaparas, Alexandros Ntoulas, and Livia Polanyi. 2010. Exploiting Social Context for Review Quality Prediction. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. Association for Computing Machinery, New York, NY, USA, 691–700.

[25] Antonis Matakos, Evimaria Terzi, and Panayiotis Tsaparas. 2017. Measuring and Moderating Opinion Polarization in Social Networks. *Data Min. Knowl. Discov.* 31, 5 (sep 2017).

[26] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring Networks of Substitutable and Complementary Products. In *KDD (KDD'15)*. ACM.

[27] Xinfan Meng and Houfeng Wang. 2009. Mining User Reviews: From Specification to Summarization. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers (ACLShort '09)*. Association for Computational Linguistics, USA, 177–180.

[28] Kazutaka Shimada, Ryosuke Tadano, and Tsutomu Endo. 2011. Multi-aspects review summarization with objective information. *Procedia - Social and Behavioral Sciences* 27 (2011), 140–149. https://doi.org/10.1016/j.sbspro.2011.10.592

[29] Panayiotis Tsaparas, Alexandros Ntoulas, and Evimaria Terzi. 2011. Selecting a Comprehensive Set of Reviews. In *KDD*. ACM.

[30] Oren Tsur and Ari Rappoport. 2009. Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 3. 154–161.

[31] Wenting Tu, David W. Cheung, and Nikos Mamoulis. 2017. More focus on what you care about: Personalized top reviews set. *Neurocomputing* 254 (2017), 3–12. https://doi.org/10.1016/j.neucom.2016.10.081

[32] Virginia Vassilevska, Ryan Williams, and Raphael Yuster. 2010. Finding Heaviest H-Subgraphs in Real Weighted Graphs, with Applications. *ACM Trans. Algorithms* 6, 3, Article 44 (jul 2010), 23 pages.

[33] Nikhita Vedula, Marcus Collins, Eugene Agichtein, and Oleg Rokhlenko. 2023. Generating Explainable Product Comparisons for Online Shopping. In *WSDM*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3539597.3570489

[34] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable Recommendation via Multi-Task Learning in Opinionated Text Data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 165–174.

[35] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z Sheng, Mehmet A Orgun, and Defu Lian. 2021. A survey on session-based recommender systems. *ACM Computing Surveys (CSUR)* 54, 7 (2021), 1–38.

[36] Nana Xu, Hongyan Liu, Jiawei Chen, Jun He, and Xiaoyong Du. 2014. Selecting a Representative Set of Diverse Quality Reviews Automatically. In *Proceedings of the 2014 SIAM International Conference on Data Mining (SDM)*. SIAM, 488–496.

[37] Aobo Yang, Nan Wang, Renqin Cai, Hongbo Deng, and Hongning Wang. 2022. Comparative Explanations of Recommendations. In *WWW*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3485447.3512031

[38] Yang Yang, Jie Tang, Jacklyne Keomany, Yanting Zhao, Juanzi Li, Ying Ding, Tian Li, and Liangwei Wang. 2012. Mining Competitive Relationships by Learning across Heterogeneous Networks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12)*. Association for Computing Machinery, New York, NY, USA, 1432–1441. https://doi.org/10.1145/2396761.2398449

[39] Wenzhe Yu, Rong Zhang, Xiaofeng He, and Chaofeng Sha. 2013. Selecting a Diversified Set of Reviews. In *Web Technologies and Applications*. 721–733.

[40] Jiaming Zhan, Han Tong Loh, and Ying Liu. 2009. Gather customer concerns from online product reviews – A text summarization approach. *Expert Syst. Appl.* 36, 2, Part 1 (2009), 2107–2115. https://doi.org/10.1016/j.eswa.2007.12.039

[41] Mingyue Zhang, Xuan Wei, Xunhua Guo, Guoqing Chen, and Qiang Wei. 2019. Identifying complements and substitutes of products: A neural network framework based on product embedding. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13, 3 (2019), 1–29.

[42] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit Factor Models for Explainable Recommendation Based on Phrase-Level Sentiment Analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '14)*. Association for Computing Machinery, New York, NY, USA, 83–92.

[43] Zhu Zhang, Chenhui Guo, and Paulo Goes. 2013. Product Comparison Networks for Competitive Analysis of Online Word-of-Mouth. *ACM Trans. Manage. Inf. Syst.* 3, 4, Article 20 (Jan. 2013).

[44] Zhu Zhang and Balaji Varadarajan. 2006. Utility Scoring of Product Reviews. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM '06)*. Association for Computing Machinery, New York, NY, USA, 51–57. https://doi.org/10.1145/1183614.1183626

[45] Di Zhu, Theodoros Lappas, and Juheng Zhang. 2018. Unsupervised tip-mining from customer reviews. *Decision Support Systems* 107 (2018), 116–124. https://doi.org/10.1016/j.dss.2018.01.011

[46] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie Review Mining and Summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM '06)*. Association for Computing Machinery, New York, NY, USA, 43–50. https://doi.org/10.1145/1183614.1183625