On Interpretation and Measurement of Soft Attributes for Recommendation

Krisztian Balog[*]
University of Stavanger
Stavanger, Norway
krisztian.balog@uis.no

Filip Radlinski
Google
London, UK
filiprad@google.com

Alexandros Karatzoglou
Google
London, UK
alexkz@google.com

ABSTRACT

We address how to robustly interpret natural language refinements (or critiques) in recommender systems. In particular, in human-human recommendation settings people frequently use *soft attributes* to express preferences about items, including concepts like the originality of a movie plot, the noisiness of a venue, or the complexity of a recipe. While binary tagging is extensively studied in the context of recommender systems, soft attributes often involve subjective and contextual aspects, which cannot be captured reliably in this way, nor be represented as objective binary truth in a knowledge base. This also adds important considerations when *measuring* soft attribute ranking. We propose a more natural representation as personalized relative statements, rather than as absolute item properties. We present novel data collection techniques and evaluation approaches, and a new public dataset. We also propose a set of scoring approaches, from unsupervised to weakly supervised to fully supervised, as a step towards interpreting and acting upon soft attribute based critiques.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Soft attributes; Recommendation critiquing; Preference Feedback

ACM Reference Format:
Krisztian Balog, Filip Radlinski, and Alexandros Karatzoglou. 2021. On Interpretation and Measurement of Soft Attributes for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3462893

1 INTRODUCTION

Recommender systems provide personalized recommendations of products or services to users, based on their preferences. A key element of recommender systems is the ability for users to provide feedback on their likes and dislikes. This is inherently a sequential interactive process, and there is a growing interest in designing automated systems that solicit users' preferences and provide personalized recommendations via multi-turn dialogue [9, 10, 21, 30, 34, 46]. Ideally, such systems should be able to incorporate any natural language user feedback on items [30].

Imagine a scenario where a conversational recommender has suggested some item. The user may respond "show me more like this but..." [27], explaining desired differences with respect to this *anchor item*. For example, they might ask for a "less violent," "more thought-provoking," or "less predictable" movie. In case of restaurants, it might be for places that are "more down home." While soft attributes are common in natural dialogue [29], interpreting them (that is, determining the degree to which they apply to items) represents the open challenge that we tackle. In contrast, most past work on attribute-based critiquing has been trained and evaluated with datasets focused on binary and unambiguous attributes (e.g., movie tags [12] or book and product categories [28, 40]).

The key difficulty stems from the unconstrained natural language of soft attributes, i.e., properties that are not universally agreed upon, or where different people have different norms, expectations, and thresholds (e.g., "exciting" vs. "boring" movies). This contrasts with well-studied objective properties like movie actors and genres, point-of-interest categories, or attributes like "plot twist." Due to their subjective nature, representing soft attributes as property-value pairs in a knowledge base is error-prone and often infeasible. Critically, while soft attributes resemble social tags, there are at least two key differences that make applying tag collections (e.g., the MovieLens Tag Genome [38]) challenging. First, tags are binary labels that do not allow for a relative ordering within relevant items (e.g., which of two movies is "more violent," if both are violent to some degree). Thus most tag-based evaluations focus on comparing items with and without a tag, being insensitive to fine-grained structure. Second, while in principle both tags and soft attributes are unconstrained natural language, most tagging approaches intentionally bias users towards a consistent vocabulary, for example displaying frequent past labels [12, 39] or pre-defining an ontology of genres [40]. Soft attributes are by definition meant to allow critiquing in natural conversations, i.e., serve a different purpose.

Specifically, this paper addresses the problem of the *interpretation* of soft attributes. Note that this is different from the task of incorporating subjective item descriptions into latent user preference representations for improving end-to-end recommendation performance (measured in terms of success rate or the number of conversational turns) [20, 24, 25, 42]. Instead, our objectives are to be able to (1) explicitly measure the degree to which a soft attribute applies to a given item and (2) quantify the degree of "softness" (i.e., subjectivity) of soft attributes.

[*] Work done while at Google.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada.

ACM ISBN 978-1-4503-8037-9/21/07.
https://doi.org/10.1145/3404835.3462893

As our first main contribution, we develop a reusable test collection, comprising a set of soft attributes and ground truth item orderings with respect to that attribute (for particular users), and an evaluation metric. The soft attributes we study come from realistic conversations of movie preferences [29]. Sampling attributes mentioned extemporaneously by participants, we develop a novel controlled multi-stage crowd labeling mechanism to collect ground truth of personalized partial orderings, while keeping workers' cognitive load low. Contrasting the attributes with tags, we note that only 47% of the soft attributes we study are present as tags in the commonly used MovieLens TagGenome [38]. To assess the quality of the scoring functions, we also propose a novel weighted extension to established rank correlation [11], based on agreement with respect to the structured ground truth ranking.

A second key contribution is explicit quantification of the subjectivity or "softness" of soft attributes. We identify ways to differentiate more from less subjective soft attributes, and for measuring how this affects item scoring. This result also has ramifications for standard tagging, drawing attention to the role of subjectivity.

As our third main contribution, we address the problem of critiquing based on soft attributes, and present empirical evidence to demonstrate the importance of debiased collection of ground truth. In particular, *soft attribute-based critiquing* is the task of ranking items *relative* to a given anchor item with respect to a given soft attribute. Three families of methods are devised for this task (unsupervised, weakly supervised, and fully supervised), which correspond to increasingly advanced approaches. These methods are compared on two test collections: one created based on existing social tags and another constructed using our proposed approach. We observe a discrepancy between the results on the two test collections, showing that the tag-based test collection is blind to item ranking improvements, making progress in this area difficult. This demonstrates how new datasets and evaluation approaches may enable significant progress in nuanced interpretation of natural language critiques, which could not previously be reliably addressed. We also analyze performance with respect to attribute "softness" and find that methods perform significantly better on attributes with higher agreement as opposed to those with low agreement.

In summary, the main novelty of the present work is the formalization of the notion of soft attributes, which opens up new possibilities for more natural interactions with conversational recommender systems. Our main technical contributions are threefold: (1) We present an efficient method for debiased collection of ground truth for comparing items with respect to a given soft attribute and release a new dataset (Section 3); (2) We introduce a measure to quantify soft attribute subjectivity, and perform an analysis of "softness" (Section 4); (3) We introduce and formalize a much refined task of critiquing based on soft attributes, developing unsupervised, weakly supervised, and fully supervised approaches (Sections 5–6).

2 RELATED WORK

We present an overview of key past work on conversational recommendation, then specifically on critiquing. Following this, we present connections to opinion mining and related item tagging.

Conversational Recommender Systems. Conversation has become recognized as a key modality for recommender systems (RSs) [19, 29, 43]. Under the broader umbrella of information seeking and recommendation, conversational interaction has been recognized as of particular interest to the research community [10]. Growing out of early work on slot filling [41] and facted search [36], conversational recommendation is distinct in that the sequence of exchanges between the user and the system is less rigid in structure, and often allow natural language dialogue. Radlinski and Craswell [30] postulated specific desirable properties of conversational search and recommendation systems in general, with critiquing being a core property. Various aspects of conversation have been addressed—including selecting preference elicitation questions to ask [9, 32], deep reinforcement learning models to understand user responses [34], multi-memory neural architectures to model preferences over attributes [46], and neural models for recommendation directly based on conversations [21]. Our work continues in this thread, albeit with a focus on semantic understanding of user utterances, at a level of detail that has not been addressed before.

Critiquing in Recommender Systems. A specific interaction in conversational recommendation, critiquing seeks user reactions to items or sets of items. As such, critiquing-based RSs make recommendations, then elicit feedback in the form of critiques [8]. The user may give feedback on various facets of importance, e.g. the airline and cost of a flight, or time and date of travel, with respect to options presented [23]. This is often repeated multiple times before the user makes a final selection. Chen and Pu [7] present a user interface facilitating critiquing, by letting users indicate how individual attributes should be changed (e.g., different screen size for a digital camera). Critiquing has also been studied in conversational RSs [27, 43] whereby users can affect recommendations along some standard item attributes with which items have been labeled, e.g., genres in movies. Bi et al. [3] present a product search model that incorporates negative feedback on certain item properties (aspect-value pairs). Unlike them, we solicit unconstrained natural language feedback, not limited to specific item properties. Critiquing also happens in human conversations, where it is also not limited to predefined facets [30]. Being able to interpret soft attributes, which is our main objective, would thus enable the development of conversational agents that more closely resemble a human-to-human conversation. Some previous work has also modelled how to learn different users' different definitions for particular terms. For instance, a *safe* car may mean different things to different people [4, 5], although evaluation was limited to simulations.

A recent line of work aims to facilitate language-based critiquing in modern embedding-based recommender systems beyond explicitly known item attributes [20, 24, 25, 42]. The central theme of these efforts is to co-embed subjective item descriptions (i.e., keyphrases from user reviews) with general user preference information. Then, end-to-end recommender architectures are trained to suppress items that match a critiqued keyphrase. We also address the problem of natural language critiquing, but our focus lies on measuring the *degree* to which a soft attribute applies to a given item, as opposed to end-to-end recommender architectures. Our ranking models also operate in a latent embedding space to help overcome data sparsity and to account for variety in language usage, but we represent soft attributes as a function over item embedding dimensions, instead of trying to co-embed them in the same space. This way we are not constrained by keyphrase extraction.

Comparative Opinion Mining. Not to be confused with opinion mining, *comparative opinion mining* deals with identifying and extracting information that is expressed in a comparative form [37]. Long studied by linguists, Jindal and Liu [13] presented the first computational approach to comparative sentence extraction. Once comparative sentences are identified, comparative elements must be extracted: the entities and the attribute/aspect on which they are being compared, the comparative predicate, and the comparison polarity. For instance, Jindal and Liu [13] identify entities by POS tagging, and recognize comparative predicates using a manually compiled list. More recently, Kessler and Kuhn [15] employ semantic role labeling techniques instead. Comparative sentences may be utilized in various ways, e.g. for determining which of two entities is better overall [18, 35, 47] or obtaining a global ranking of entities on a given aspect [22]. A typical approach is to build a directed graph of entities, where edge weights encode the degree of belief that one entity is better than the other on a given aspect. Then, entities can be ranked by some measure of graph centrality [18, 22].

However, the aspects considered previously are limited, often coming from a fixed ontology (e.g., [22]). Further, comparative opinion statements in natural text are uncommon, with estimates that 10% of sentences in typical reviews contain a comparison [16]. The most important difference in our work is that we aim to interpret arbitrary critiques, allowing direct navigation of the recommendation space. By designing a data collection and evaluation specifically for this task, we do not limit ourselves to common terms nor to reviews, but rather to natural critiques.

3 BUILDING A TEST COLLECTION FOR CRITIQUING BASED ON SOFT ATTRIBUTES

We begin this section by motivating and formally introducing the notion of *soft attributes* (Sec. 3.1). Next, we describe the process of creating a reusable test collection for determining how a soft attribute applies to given items. At a high level, the process consists of (1) sampling soft attributes, (2) obtaining ground truth item orderings for each attributes, and (3) developing an appropriate evaluation measure for this task. A main challenge is that obtaining complete and universal ground truth orderings of items is infeasible, because of the human annotation effort required, and because people often disagree on the relative ordering of items. We therefore use two types of evaluation collections. The first, which we term the MovieLens Attribute Collection, reuses past social tag annotations, with all their pertinent limitations and biases (Section 3.3). Our second dataset, which we term the Soft Attribute Collection, is collected with a process designed to reduce the biases (Section 3.4).

3.1 Defining Soft Attributes

Consider the partial exchange between a user and an agent shown in Table 1. The terminology employed and the aspects of this particular movie that attract the user's interest are likely to be useful for an RS providing future recommendations. Here, the user mentions concepts such as the *immersiveness* of the movie, the *in-depth exploration of a particular time period*, a *bumbly character* that they relate to, and the *level of violence* depicted. Yet it is difficult to argue that many of these attributes could be attached as definitive labels (i.e., tags) for the movie. Rather, the concepts are *soft* attributes.

Table 1: Example partial exchange between a user and an agent about the user's movies interests [29].

USER	Zodiac's one of my favorite movies.
USER	Zodiac the movie about a serial killer from the '60s or '70s, around there.
ASSISTANT	Zodiac? Oh wow ok, what do you like about that movie?
USER	And I just think serial killers, in general, are interesting, so the movie was really good. And it was just neat to see that world. Like it went really in-depth. It was like almost 3 hours long, so you got to really feel like you were a part of that world and time period. And see what the detectives and the police did and the investigators.
ASSISTANT	So you feel like you were part of that world ?
USER	Yeah. It was really an immersive movie.
ASSISTANT	If you were in the movie what character do you feel most relatable
USER	Probably the main character, Robert Graysmith, just cuz he's awkward and bumbly. So, I guess that's the one I would be the closest to.
ASSISTANT	What scene do you like the best?
USER	Probably the most memorable one is the murder at the lake. Just cuz it's really vivid and horrific to watch. But it's very memorable.

Definition. *We define a **soft attribute** as a property of an item that is not a verifiable fact that can be universally agreed upon, and where it is meaningful to compare two items and say that one item has more of the attribute than another.*

A number of examples of soft attributes are provided in Table 2 (Soft Attribute Collection). We highlight two main distinguishing features that make a given attribute *soft*. First, there is commonly a question of degree. For instance, consider the movie attribute "violent." If taken to either apply or not apply to a particular movie, it would be natural to ask what it takes for a movie to be marked as such. Are there different types of violence? How much violence is necessary in a movie before an entire movie is considered violent? Thus it is critical to model the degree to which each soft attribute applies to a given item. Second, people may disagree in their assessment, even with a real-valued measure. While there are societal norms for attributes like violence (reflected in movie rating criteria, corresponding to a rating scale), this need not apply in general. In fact, we may ask whose opinion about an attribute (such as "violence") should matter if a given user says that they prefer "less violent movies"? As such, different people may have different norms, expectations and thresholds for a given soft attribute. Boutilier et al. [4, 5] for instance postulated that an attribute such as "safety" may have different sub-aspects, with different people placing different importance on these sub-aspects. We also include soft attributes that do not have norms: for example, "exciting" and "boring" are attributes where different people may entirely disagree.

Another important difference of soft attributes versus social tags is that most tagging approaches intentionally bias users towards a consistent vocabulary, for example displaying frequent past labels [12, 39] or pre-defining an ontology of genres [40]. This encourages social taggers to conflate different semantics in popular tags.

We also note that soft attributes do not need to necessarily apply to all items in the collection. For example, "realistic CGI" in a movie is a soft attribute, yet CGI is not always present. This means that for any given soft attributes there is a (personal) partial order over items, where some items have the attribute more or less than others, while others are incomparable.

Table 2: Examples of soft attributes from our test collections.

MovieLens Attribute Collection			
action	comedy	depressing	dystopia
romance	sci-fi	superhero	suspense
Soft Attribute Collection			
artsy	feels real	incomprehensible	intense
light-hearted	predictable	simplistic script	violent

3.2 Items Considered

Here, we work within the movies domain as it has received the most attention in recommender systems research, and has a rich variety of publicly available resources (e.g., [12, 29]). However, we note that there is nothing domain dependent in our approach and the same procedure may be employed to create test collections for other domains. Our corpus consists of the 300 most popular movies in the MovieLens-20M collection [12], where popularity is measured by the number of users who rated the movie. We annotate each movie with its reviews in the Amazon review corpus [26], using the mappings established in [44].[1]

3.3 MovieLens Attribute Collection

As frequently done [6, 31, 33], one can create a synthetic test collection based on social tags (such as those present in MovieLens). We do so, noting that its main limitation is that it does not allow for to measurement of the "degree" of a soft attribute, only its presence or absence. We shall investigate later on (in Section 6) how this affects the observed performance of ranking methods.

3.3.1 Soft Attributes. The top 100 most frequent MovieLens tags are taken as soft attributes, excluding tags that are named entities (names of actors, directors, studios, etc.), refer to adult content, or contain coarse language. We note that each of the tags has been assigned to 200–900 different items. Table 2 shows examples.

3.3.2 Ground Truth. We identify two sets of items: those that have not been assigned that tag by any user, $\mathcal{X}^-$, and those where a significant portion α of users who assigned any tag to an item assigned the given tag, $\mathcal{X}^+$. We take the threshold to be $\alpha = 0.15$. This is a conservative threshold, leaving typically only a handful of items as positive examples for each tag. Arguably, more items could be included, but that would come at the expense of potentially introducing noise. A manual inspection of the results showed that all items in $\mathcal{X}^+$ appear to be valid examples for the tag. Because of this conservative threshold, there were no positive examples for 18 tags, leaving us with 82 tags as soft attributes.

For reliable statistics, we also require items to be tagged by at least 50 users. This leaves 238 out of the original 300 movies in this test collection. Table 3 presents descriptive statistics for the dataset.

3.3.3 Evaluation Measure. To be able to evaluate algorithms that score items with respect to soft attributes, we need a measure that defines the goodness of a ranking. Our evaluation measure is based on the number of concordant and discordant pairs between the generated ranking and the ground truth, specifically Goodman and Kruskal's gamma [11] as a measure of rank correlation:

Table 3: Descriptive statistics of the test collections.

MovieLens Attribute Collection	
Description	Count (min/mean/max)
Number of soft attributes	82
Positive examples / attribute	1 / 5.77 / 28
Negative examples / attribute	144 / 214.7 / 234
Soft Attribute Collection	
Description	Count
Number of soft attributes	60
Sets of movies rated	5,991
Pairwise preferences	249,863 with 52,352 ties

$$G = \frac{N_s - N_d}{N_s + N_d}\ , \tag{1}$$

where N_s is the number of concordant pairs and N_d is the number of discordant pairs with respect to the ground truth ordering. G can range from −1 (perfect inversion) to +1 (perfect agreement).

3.4 Soft Attribute Collection

While the MovieLens Attribute Collection was created using a common approach, tags suffer from a number of biases. Not least, item popularity biases the potential for items to receive particular tags, and the very nature of tagging limits the set of assigned tags to a handful of most strongly associated ones. Yet, the fact that a tag has not been assigned to an item does not necessarily mean that it does not apply to that item at all. Here, we present a multi-step approach that aims to reduce such biases. The final dataset statistics are presented in Table 3, with the data available for download.[2]

3.4.1 Soft Attributes. We sample soft attributes from the CCPE-M dataset [29]. It consists of over 500 English dialogs between a user and an assistant discussing movie preferences in natural language. Utterances are annotated with (1) entity mentions, (2) preferences about entities, (3) descriptions of entities, and (4) other statements of entities. Of these, we leverage annotations of preferences (2). First, we filter for utterances that potentially express a preference, by simply checking for the presence of the words "more," "less," or "too." The resulting utterances were evaluated manually, and soft attributes identified. A conservative filtering was applied, removing expressions that were problematic even for humans to interpret. This yielded 173 unique soft attributes, of which we sample 60 for our evaluation (the probability of selecting an attribute was proportional to its frequency of use). Table 2 lists a few examples.

3.4.2 Ground Truth. Our methodology for collecting ground truth for soft attributes is one of the key contributions of this work. In particular, we focus on efficient collection of relative preferences over pairs of items for the same soft attribute, to enable inter-rater effects to be fairly measured. Additionally, the rater interface is designed to reduce popularity biases, as well as biases due to certain soft attributes never even being considered for certain items.

[1] http://goo.gle/research-docent

[2] http://goo.gle/soft-attribute-data

Annotation Procedure. For these attributes, we design a two-stage procedure to obtain pairwise item orderings via crowdsourcing. In particular, we recruited native English speakers based in the United States and Canada with a record of high quality annotation.

In Stage 1, the crowd workers are asked to indicate which movies they have seen, from a pool of popular movies (cf. Section 3.2).

In Stage 2, each worker is presented with a specific ordering task for each soft attribute a. As illustrated in Figure 1, they are provided with an anchor item x and a small sample set of items (usually 10), which are to be divided into three categories with reference to the anchor item: "less a than x," "about the same a as x," and "more a than x." We represent each item by its image (movie poster) and title. The set of items shown (both the anchor item and the sample set) are personalized for each worker, that is, they are chosen from the set of movies that the person has indicated as seen in Stage 1 so as to ensure that all judgments compare items on an equal footing. By forcing workers to consider consistent terminology across a variety of items, we reduce biases due to the context in which individual items are usually considered, and typical comparisons which are made (e.g., dominated by the genres to which items belong, or tags given by other users in the past). By offering the "about the same" label, we also force an explicit decision as to whether a difference between two items is meaningful or not. This full set of judgments then yields pairwise preferences over all pairs of items placed into two different classes, as well as relative to the anchor item.

Sampling Methodology. Notice that the way items are selected for crowd workers influences the number of pairwise item preferences that can be obtained. To maximize this value of the rating task, we develop the following sampling methodology. To increase the likelihood of meaningful comparisons between items, we score items using two baseline algorithms: the term-based item-centric and review-centric models (described in Section 5.2.1). First, we eliminate all items that the user has not marked as seen in Stage 1. All seen items are then sorted by score and partitioned into $M = 5$ bins for each baseline, each with about the same number of items.

For a particular user and attribute a, we must select the anchor item x and subset $\mathcal{X}$ to rank. The choice of anchor item x is particularly key: if it is at an extreme for the soft attribute, most items that x is compared to would fall into just one or two buckets, yielding few pairwise preferences. Therefore, only items not in the first or last bin for either baseline shall be selected as x. Second, as the anchor item is present in the most pairwise preferences inferred, we wish to bias it towards more popular items, maximizing the number of observed preferences for this attribute for different workers. Thus, from the set of possible anchor items, we sample the anchor item with probability proportional to the number of crowd workers who indicated that they had seen this item in Stage 1. Finally, $\mathcal{X}$ is a stratified sample of one item from each of the M bins for each baseline algorithm, sampling items with probability proportional to the number of workers who marked each item as seen in Stage 1. This gives $|\mathcal{X}| = 10$ items, unless the user has seen too few items.

3.4.3 Evaluation Measure.

As explained above and in Figure 1, raters (crowd workers) are presented, for each soft attribute, with a reference item x and a sample set of items $\mathcal{X}$, to be categorized into three disjoint sets: $\mathcal{X}^-$ ("less a than x"), $\mathcal{X}^\circ$ ("about the same a as x"), and $\mathcal{X}^+$ ("more a than x"). A standard approach for evaluating scoring algorithms given pairwise preferences is to count how many pairs are ordered in agreement with the test data.

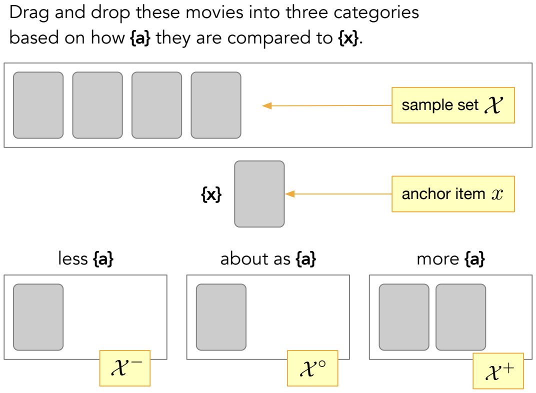

Figure 1: User interface for obtaining item orderings. Given a soft attribute (a) and a set of items ($\mathcal{X}$), workers are required to drag the image for each item based on how this item compares to the anchor item (x) with respect to a. Items are thus partitioned into three classes, $\mathcal{X}^-, \mathcal{X}^\circ$ and $\mathcal{X}^+$.

However, with three classes, we are able to measure agreement with a stronger metric. We present an extended version of the pairwise agreement metric that differentiates between more/less and *much* more/less. Specifically, we define a weighted extension of Goodman and Kruskal's gamma rank correlation measure [11]:

$$G' = \frac{(N_s - N_d) + 2(N_{ss} - N_{dd})}{(N_s + N_d) + 2(N_{ss} + N_{dd})}, \qquad (2)$$

where the number of concordant and discordant pairs, N_s and N_d, are measured on elements in adjacent buckets, that is, $\mathcal{X}^-$ vs. $\mathcal{X}^0 \cup \{x\}$ and in $\mathcal{X}^0 \cup \{x\}$ vs. $\mathcal{X}^+$ (the anchor item x is considered in the "middle" bucket).[3]

For pairs of items judged *much* more/less (in buckets $\mathcal{X}^-$ vs. $\mathcal{X}^+$), we give double weight: N_{ss} counts the number of pairs of items in $\mathcal{X}^-$ vs. $\mathcal{X}^+$ that are ranked identically, while N_{dd} counts how many of those are ranked differently, in a given item ranking as compared to the ground truth. Similar to the original G (Eq. (1)), the extended G' also ranges from -1 (perfect anti-correlation) to +1 (perfect correlation), with a score of 0 indicating random agreement.

In the rare case where all items are rated into a single bucket, namely to "about the same a as x," this metric is undefined. We leave out these ratings from the evaluation.

Note that unlike the original G, our extended G' does not assume an universally accepted ground truth. Instead, it measures how well a ranking aligns with the preferences of a single rater. In our experiments, we report on mean G' values over all raters.

4 ANALYSIS OF SOFT ATTRIBUTES

There are two particularly fundamental concepts when considering a given soft attribute: (1) How subjective is the meaning of the *text* used for that attribute when applied to particular items?, and (2) Do we observe significant "equality?" That is, when a user requests an item with a given attribute changed, how much change is expected? We address what our data tells us about each in turn.

[3] Measurement on elements in buckets $\mathcal{X}^a$ vs. $\mathcal{X}^b$ means that all pairs of items $x_a \in \mathcal{X}^a, x_b \in \mathcal{X}^b$ are considered; if x_a is ranked higher (i.e., given a higher score) than x_b in the ranking, then it counts as a concordant pair, otherwise as a discordant pair.

4.1 Quantifying Subjectivity

Past attribute datasets have not analyzed personal variations in the *meaning* of a term, or in the *relative applicability* of soft attributes: While often scored for relevance (as in [39]), it is difficult to argue that scores are an unbiased measure of disagreement about meaning. In our work, these questions are reflected in inter-judge agreement: Do different people considering the same soft attribute for the same items agree on which is "more"?

For example, consider the attribute "predictable." Different people may consider different plot devices predictable, based on their past experience, leading to disagreement about the relative ordering of some movies. On the other hand, we might consider the attribute "funny" under-specified: for some people it may indicate slapstick comedy, while for others it may indicate peculiar characters, plots, and settings (often characteristic of British comedy). As such, disagreements may come from different definitions.

To measure soft attribute subjectivity, we identify all pairs of movies (x, x') that have been ranked for the same attribute a by different raters. At this point, recall that multiple judgment of tuples (a, x, x') is not something we can directly control. Rather, raters specify which movies they have seen, and anchor items and movies are sampled as described above—meaning we rely on this being an unbiased sample of such tuples. Most tuples were judged just twice.

We define preference agreement by considering every pair of judgments for every soft attribute. Agreement is defined as the fraction of pairs where either (1) both raters agree on the direction of preference, or (2) at least one rater indicates a lack of preference (i.e., there is no disagreement). Formally, let $N^a_{x<x'}$ be the number of raters who indicated that attribute a applies more to x' than to x, and $N^a_{x\approx x'}$ that for raters who indicated that a applies about equally. For each pair of movies x and x',

$$p^a_{x<x'} = \frac{N^a_{x<x'}}{N^a_{x<x'}+N^a_{x=x'}+N^a_{x>x'}} \quad p^a_{x\approx x'} = \frac{N^a_{x\approx x'}}{N^a_{x<x'}+N^a_{x=x'}+N^a_{x>x'}} \\ p^a_{x>x'} = \frac{N^a_{x>x'}}{N^a_{x<x'}+N^a_{x=x'}+N^a_{x>x'}} \tag{3}$$

For attribute a, $agree(a) = mean_{x,x'}\left[agree(a, x, x')\right]$, where

$$\begin{aligned} agree(a, x, x') \quad = \quad & p^a_{x<x'} \cdot \left(p^a_{x<x'} + p^a_{x\approx x'}\right) \\ & + p^a_{x>x'} \cdot \left(p^a_{x>x'} + p^a_{x\approx x'}\right) + p^a_{x\approx x'}, \end{aligned} \tag{4}$$

taking lack of preference ($x \approx x'$) as indicating agreement with either direction. Based on agreement rate, we divide the soft attributes into three equal-sized groups: High, Medium, and Low agreement attributes. A sample from each is shown in Table 4. We see that the attributes with highest score are reminiscent of typical tags in tag corpora for movies (we found it surprising that *funny* has high agreement). Many of the attributes with lowest agreement relate to personal preferences (*entertaining*, *overrated*). It is also particularly interesting to observe that attributes that are seemingly opposite (e.g., *intense* and *boring*) can have quite different agreement rates.

4.2 Quantifying Equality

As we tasked raters to bucket movies into three categories, we can observe the distribution over the counts in these buckets. Given an anchor item and attribute, on average raters put[4] 3.46 movies into $\mathcal{X}^-$, 3.28 movies into $\mathcal{X}^\circ$, and 3.19 movies into $\mathcal{X}^+$. While this distribution is likely influenced by the stratified sampling of items in $\mathcal{X}$, it confirms that there often exist many pairs of movies to which a given attribute applies equally—even when many others can be classified as more or less. Considering the scores of movies in the $\mathcal{X}^\circ$ set ("about the same a as x"), it would be possible to identify thresholds that determine how different scores for items should be for a recommender to have satisfied a user's critique of more/less without necessitating extreme differences. For instance, asking for a "less violent" movie compared to a very violent one should likely not lead to recommendations of movies with no violence at all.

Table 4: Selection of attributes with High, Medium, and Low agreement rates (*agree*(a)).

Group	Attribute	$agree(a)$	Distinct movie pairs	Total comparisons	
High agreement	scary	0.962	291	617	(6 ties)
	gory	0.952	246	513	(3 ties)
	action filled	0.950	277	583	(23 ties)
	funny	0.949	318	672	(10 ties)
	terrifying	0.947	289	623	(19 ties)
	violent	0.946	290	615	(13 ties)
	intense	0.937	297	634	(27 ties)
Medium agreement	fictionalized	0.894	243	511	(12 ties)
	tearful	0.880	233	493	(10 ties)
	romantic	0.885	272	585	(21 ties)
	confusing	0.882	183	390	(16 ties)
	mushy mushy	0.855	241	510	(16 ties)
	exaggerated	0.830	288	606	(24 ties)
Low agreement	unique story	0.813	225	470	(22 ties)
	original	0.808	189	391	(20 ties)
	entertaining	0.796	270	569	(18 ties)
	boring	0.791	234	514	(28 ties)
	dynamic	0.785	280	590	(25 ties)
	overrated	0.766	280	596	(26 ties)

We also observe that *long* (4.55 movies in $\mathcal{X}^\circ$), *documentary style* (4.51 movies), *well directed* (4.13 movies) and *original* (4.13 movies) have the most items in $\mathcal{X}^\circ$. This suggests that critiques of these attributes are likely to eliminate many movies simply because significant differences are less common. In contrast, the attributes *playful* (2.53 movies in $\mathcal{X}^\circ$), *funny* (2.57 movies), *sappy* (2.69 movies) and *scary* (2.78 movies) are much more likely to have raters provide a more complete order over movies, thus fewer would be eliminated on the grounds of being too similar so as to satisfy a critique.

5 SCORING ITEMS BY SOFT ATTRIBUTE

We now address the question of how to apply soft attribute critiques in a recommendation setting. Specifically, we cast this task as a ranking problem. Our approach hinges on the idea that for the type of critiquing we aim to support—specifying some soft attribute of the desired item(s) relative to a given anchor item—it is sufficient to determine the relative ordering of items. Once that relative ordering is established, we can locate the anchor item in the ranking and move in the required direction (and distance) with respect to that (e.g., "little less" or "much more"). Our formal objective then is to devise a scoring function $score(x, a)$, where x is an item in the collection and a is a soft attribute. If an attribute cannot be applied at all to a given item, it should ideally be assigned a score of 0.

At their core, the scoring functions boil down to the question of how items and soft attributes are represented. We represent items in a latent space by learning embeddings from item-rating data (Sec. 5.1). Our underlying assumption is that a representation of

[4] These counts do not add up to 10 as $|\mathcal{X}|$ was not always 10.

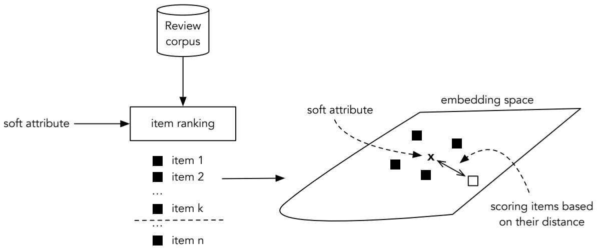

Figure 2: Unsupervised item ranking using the Centroid-Based (CB) method.

soft attributes within the same embedding space can be learned. (Note that we do not assume interpretability of these embeddings.) We learn to represent soft attributes in three fundamentally different ways, from *unsupervised* to *weakly supervised* to *fully supervised.*

- When no explicit training data over soft attributes exists, we use implicit signals from item reviews, employing established models from entity retrieval. Taking the centroid of top-ranked items, in the spirit of pseudo-relevance feedback, we representation soft attribute using an *unsupervised* approach (Section 5.2).
- Instead of considering only positive examples, we can also take the top-ranked and bottom-ranked items as positive and negative (pseudo-)training examples, respectively, to learn a pointwise regression model in a *weakly supervised* way (Section 5.3).
- Finally, where pairwise item orderings for a given soft attribute are available (facilitated by the data collection methodology we introduce), we can use these pairwise preferences as training examples and learn a model using *full supervision* (Section 5.4).

We note that these models do not personalize the meaning of a soft attribute to a particular user, although we analyze the importance of personalization in Section 6.3.2.

5.1 Generating Item Embeddings

Many methods have been proposed to compute item embeddings, e.g., [2]. One of the most common ways to compute item representations from collaborative filtering datasets is matrix factorization [17]. Matrix factorization for collaborative filtering works by using the user-item rating matrix $\mathbf{R}^{n \times m}$ for n users and m items, factorizing this matrix to two low rank matrices containing the user embeddings $\mathbf{U}^{n \times d}$ and item embeddings $\mathbf{X}^{m \times d}$. To this end the following objective function is minimized:

$$(r_{ij} - \langle \mathbf{u}_i, \mathbf{x}_j \rangle)^2 + \lambda_1 \|\mathbf{u}_i\|^2 + \lambda_2 \|\mathbf{x}_j\|^2 \; , \tag{5}$$

where r_{ij} is the rating of item j by user i, $\mathbf{u}_i$ is the d-dimensional embedding for user i and $\mathbf{x}_j$ that for item j. Each user and item embedding vector corresponds to one row in the user and item embedding matrices. The objective function is minimized using stochastic gradient descent.

5.2 Unsupervised Ranking

We present two unsupervised ranking approaches as baselines: one operating in the term space (Section 5.2.1) and another operating in the embedding space (Section 5.2.2).

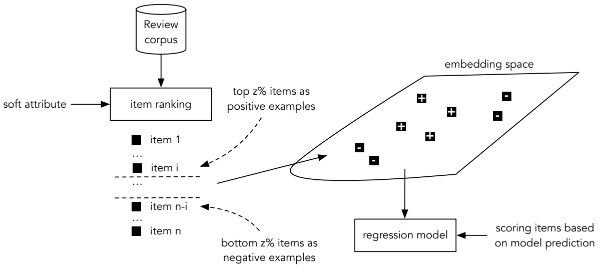
Review corpus
soft attribute
item ranking
top z% items as positive examples
embedding space
item 1
...
item i
...
item n-i
...
item n
bottom z% items as negative examples
regression model
scoring items based on model prediction

Figure 3: Weakly supervised item ranking using the Weakly-supervised Weighted Dimensions (WWD) method.

5.2.1 Term-based Ranking. Adapting well-established evidence aggregation strategies for entity retrieval [1, 45], we leverage the corpus of item reviews, using soft attributes as search queries, for our term-based (TB) methods. Items are represented by aggregating reviews following an *item-centric* or a *review-centric* strategy.

Item-centric Method A term-based representation d_x is built for each item x by concatenating all reviews mentioning x: $d_x = \bigcup_{r \in R_x} r$. The representations are then scored using standard text-based retrieval models (here: BM25):

$$score^{item}(x, a) = score_{BM25}(d_x, q) \; . \tag{6}$$

Review-centric Method Reviews $r \in \mathcal{R}$ are ranked using standard retrieval models (here: BM25). Then, for each item x, the retrieval scores of reviews that mention x are aggregated:

$$score^{review}(x, a) = \sum_{r \in R_x} score_{BM25}(r, q) \; , \tag{7}$$

where R_x denotes the set of reviews that mention x.

5.2.2 Centroid-based Ranking. Our centroid-based (CB) ranking method considers the top ranked items, in the spirit of pseudo relevance feedback, as representative examples of the soft attribute. Then, the centroid of these top ranked items is taken as the representation of the soft attribute in the embedding space. Other items can then be scored by computing their distance to the centroid.

Let $\mathbf{v}_a$ denote the representation of the soft attribute in the item embedding space (an N-dimensional vector). We obtain $\mathbf{v}_a$ by first ranking items with respect to the soft attribute using a term-based model over review text, and then taking the centroid of the top-k ranked items' embeddings:

$$\mathbf{v}_a[i] = \frac{1}{k} \sum_{x \in X_k} \mathbf{x}[i] \; , \tag{8}$$

where X_k denote the set of top-k ranked items and $\mathbf{x}[i]$ refers to the i^{th} dimension of embedding vector $\mathbf{x}$. Once $\mathbf{v}_a$ has been computed, a given item x in the collection is scored against it by computing the cosine similarity of their respective embedding vectors: $score^{CB}(x, a) = cosine(\mathbf{x}, \mathbf{a})$. See Fig. 2 for an illustration.

5.3 Weakly Supervised Ranking

While the centroid-based method represents items in the embedding space, it considers all factors (dimensions) equally important. Instead, we would like to *learn* which factors in the embedding space encode a particular soft attribute. In the absence of explicit training labels, we again use term-based models for obtaining an initial ranking of items. Then, the top- and bottom-ranked items are taken as positive and negative training examples, respectively, to learn a regression model. Items can then be scored by applying this model and taking the prediction probabilities as scores.

Specifically, we employ Logistic Regression (LR), with target labels y_a, where $y_a = 1$ when attribute a applies to item x, and $y_a = 0$ otherwise. To obtain these labels for training the LR model, we operationalize the notion of an attribute applying to an item by considering the item's position in the ranking for that attribute in the term-based model (thereby essentially taking the term-based ranking as a weak supervision signal). Items in top $z\%$ positions of the ranking are taken as positive examples (as long as their initial scores are greater than zero). The same number of items from the bottom are taken as negative. The embedding of the soft attribute can then be computed by fitting a LR of the item embeddings $\mathbf{x}$ to the item attribute labels y_a (see Fig. 3 for an illustration):

$$p\left(y_a = 1|x\right) = \frac{1}{1 + \exp\left(-\mathbf{w}_a \cdot \mathbf{x}\right)} \tag{9}$$

Intuitively, the model parameters $\mathbf{w}_a$ that are computed by LR reflect the importance (weight) of each dimension in the item embeddings in predicting the soft attribute. In this way, we compute which dimensions (and to what degree) are encoding this attribute. This can be seen as distilling a representation of the soft attribute from the item embeddings. Naturally one can compute the scores of items $\mathbf{x}$ with respect to the attributes by using the corresponding LR model. Our *Weakly-supervised Weighted Dimensions* (WWD) method thus ranks items according to:

$$score^{WWD}(x, a) = p\left(y_a = 1|x\right) \tag{10}$$

5.4 Fully Supervised Ranking

The weakly supervised method (WWD) learns weights over the embedding space dimensions using pseudo-relevance feedback based on the *review* corpus. Here, we show how the explicit item orderings (i.e., data collected using the methodology we describe in Section 3.4) can be leveraged to efficiently learn this weighting directly from the *ordinal labelling* produced by raters using a ranking algorithm trained on pairwise preferences.

Specifically, given ground truth judgments as described in Section 3.4.2, we infer all possible pairwise preferences. For each judgment with reference item r and labelled sets $\mathcal{X}^-$, $\mathcal{X}^\circ$, and $\mathcal{X}^+$ we infer the following sets of preferences between items i and j:

(1) $\{i \gg j\}$, $\forall i \in \mathcal{X}^+, j \in \mathcal{X}^-$
(2) $\{i \succ j\}$, $\forall i \in \mathcal{X}^+, j \in \mathcal{X}^\circ \bigcup \{r\}$
(3) $\{i \succ j\}$, $\forall i \in \mathcal{X}^\circ \bigcup \{r\}, j \in \mathcal{X}^-$
(4) $\{i \sim j\}$, $\forall i, j \in \mathcal{X}^\circ \bigcup \{r\}$

The fully supervised model (*Supervised Weighed Dimensions*, SWD) then uses a linear ranking support vector machine[5] [14]. Briefly, each preference p was transformed into a constraint: $i \succ j$ becomes $score^{SWD}(x_i) \geq 1 + score^{SWD}(x_j) - \xi_p$, and $i \gg j$ becomes $score^{SWD}(x_i) \geq 2 + score^{SWD}(x_j) - \xi_p$, where x_i denotes the embedding vector of item i. Then $score^{SWD}(x_i) = w \cdot x_i$, where w is learned by minimizing $\frac{1}{2} w \cdot w + C \sum_p \xi_p$.

Table 5: Soft attribute ranking results on the MovieLens Attribute Collection (MovieLens) and Soft Attributes Collection (SoftAttr). The best scores in each block are in bold.

Method		MovieLens (G)	SoftAttr (G')
Unsupervised methods			
(TB-IC)	Term-based, item-centric	**0.800**	0.110
(TB-RC)	Term-based, review-centric	0.733	**0.136**
(CB+TB-IC)	Centroid-based (w/ TB-IC)	0.404	0.087
(CB+TB-RC)	Centroid-based (w/ TB-RC)	0.471	0.101
Weakly supervised methods			
(WWD+TB-IC)	Weakly-sup. w. dim. (w/ TB-IC)	**0.539**	0.194
(WWD+TB-RC)	Weakly-sup. w. dim. (w/ TB-RC)	0.517	**0.200**
Fully supervised method			
(SWD)	Supervised weighted dimensions		**0.485**

As with the WWD model, this model represents a soft attribute by a direction in the embedding space (w), albeit trained using explicit pairwise preferences between items rather than based on terms extracted from reviews.

6 EVALUATION

We evaluate scoring algorithms based on how many pairs are ordered in agreement with the ground truth data. Specifically, we use the original Goodman and Kruskal gamma (G, Eq. (1)) for the MovieLens Attribute Collection (MovieLens for short) and our modified version (G', Eq. (2)) for the Soft Attributes Collection (SoftAttr for short). Our focus is to verify whether the algorithmic improvements we would expect are indeed observed when they are evaluated against different test collections.

6.1 Experimental Setup

For the unsupervised (CB) and weakly-supervised (WWD) methods, we report on a single parameter setting, selecting that which performs best across the board. Specifically, we use 25 dimensional embeddings; for the centroid-based methods we use top $k = 5$ items; for weighted-dimensions we take $z = 0.4$.[6]

For the supervised method, SWD, we use 10-fold cross validation: We split the raters into 10 groups, and evaluate each group by using only the remaining 9 groups to train the ranking model. We report average performance across the 10 folds.

6.2 Results

Our overall results are reported in Table 5, averaged over all soft attributes and all users. The best performing methods in the supervised and unsupervised categories are bolded for each collection.

[5] The hyperparameter C was set to 1. We analyzed the effect of tuning hyperparameters and using non-linear models including gradient boosted decision trees, finding no meaningful improvements. We hypothesize this lack of sensitivity to the precise ranking model to be due to the limited amount of training data, leaving identifying the best performing model to future work as it is not key to this work.

[6] We also performed a sensitivity analysis of these parameters and observed stable performance across a broad range of values ($k \in [1..100], z \in [0.1 .. 0.4]$). Additionally, we experimented with the dimensionality of embeddings (between 25 and 100), finding that higher dimensionality somewhat reduces performance due to over-fitting effects that frequently manifest themselves in the outright performance of high-dimensional matrix factorization techniques fitted on (relatively) small data.

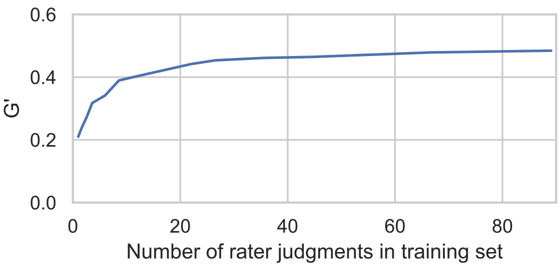

Figure 4: SWD model performance (all soft attributes averaged) vs. the number of rater's judgments used for training.

First, observe the *magnitude* of the rank correlation scores when evaluated on each collection. The simple term-based models perform remarkably well when assessed on MovieLens, to the extent that one could assume the problem of ranking items for a given soft attribute could be substantially solved with a straightforward model. Yet we postulate that this common formulation captured by the MovieLens collection, namely with performance determined by how well items *with* a given tag can be distinguished from those *without*, is misleading. On the SoftAttr collection we see much lower scores overall, indicating that this more accurate abstraction of the attribute ranking problem proves to be considerably harder.

Next we observe that the relative ordering of systems on the MovieLens collection is TB > WWD+TB > CBE+TB, while on SoftAttr collection it is WWD+TB > TB > CBE+TB, also emphasizing the importance of the task encoded in the data enabling meaningful progress. Centroid-based ranking (CB+TB) is least performant in both cases. However, the retrieval-based baseline (TB) performs much better on the MovieLens collection than the weakly supervised approach (WWD+TB). On the SoftAttr collection it is exactly the other way around. We hypothesize that one of the main reasons that WWD+TB performs much better than CB+TB is that WWD+TB considers positive as well as negative evidence (i.e., movies at the bottom of the ranking for a given attribute), while the CB+TB baseline only considers positive evidence (i.e., movies where the soft attribute can reliably be considered to apply).

Finally, we turn to the supervised method, Supervised Weighted Dimensions (SWD). We see that direct supervision based on relative preference judgments yields significant improvement in performance on the SoftAttr collection.

Together, these results show the key contribution that our new data collection methodology makes to enable the soft attribute ranking problem to be addressed.

6.3 Analysis

Given that the fully supervised SWD algorithm significantly outperforms the unsupervised and weakly supervised methods on the Soft Attribute collection, we now analyze this model further by asking: How much rater training data is required to effectively learn a ranking function for a soft attribute (Sec. 6.3.1)? And, how does model performance change across different soft attributes (Sec. 6.3.2)?

6.3.1 Data Efficiency. Figure 4 shows average SWD performance as a function of the number of rater's judgment (sets) provided to the learning algorithm. Specifically, we subsample raters in the training set and compute mean G', averaging over 25 runs at each subsample size. We see that the model is very data efficient: For any given soft attribute, judgments from approx. 20 raters are required to obtain near-optimal performance. This reinforces the value of pairwise preferences over a controlled sample of known items, as opposed to relying on binary judgments of tag presence or absence.

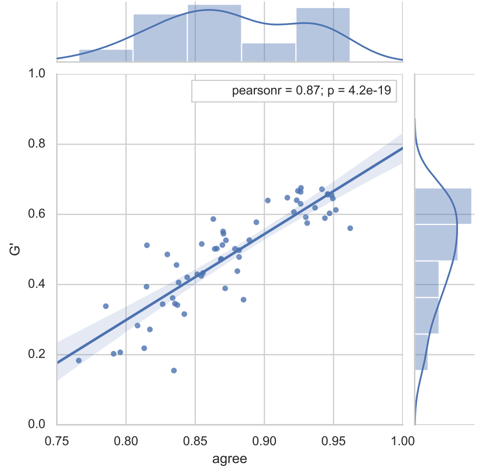

Figure 5: SWD model performance (G') against attribute subjectivity (agree). Each point represents one soft attribute.

6.3.2 Performance Analysis by Soft Attribute. Next, we plot model performance against attribute subjectivity in Figure 5. We observe a clear correlation between the subjectiveness of a soft attribute (measured in terms of inter-rater agreement) and ranking performance (in terms of weighted gamma rank correlation G'). As may be expected, soft attributes with less agreement are harder to predict, leaving significant room for personalized soft attribute scoring models as future work. These results also suggest that predicting the "softness" of a soft attribute is also an important future direction.

7 CONCLUSIONS

In this work, we have formalized recommender system critiquing based on *soft attributes*, that is, aspects of items that cannot be established as universally agreed facts. Developing a general methodology for obtaining soft attribute judgments, we have presented a dataset of pairwise preferences over soft attributes applied to the domain of movies. Our analysis of these preferences has shown that different attributes exhibit different levels of "softness," with those that have less inter-user agreement being more difficult to model. We have also developed approaches for critiquing based on soft attributes, in particular introducing a set of scoring approaches, from unsupervised to weakly supervised to fully supervised, employing both information retrieval approaches and embedding models. Moreover, comparisons on a standard tagging dataset has demonstrated how an appropriate data collection approach is key to making progress on the soft attribute scoring task.

Personalization is a key aspect of soft-attribute ranking that was not addressed in this paper, despite a clear signal of its importance. In fact, our analysis of performance by soft attribute suggests significant headroom for personalized soft attribute ranking methods as future work: The results suggest that "softer" soft attributes perform more poorly with non-personalized algorithms.

REFERENCES

[1] Krisztian Balog. 2018. *Entity-Oriented Search*. The Information Retrieval Series, Vol. 39. Springer.

[2] Oren Barkan and Noam Koenigstein. 2016. Item2Vec: Neural Item Embedding for Collaborative Filtering. *CoRR* abs/1603.04259 (2016).

[3] Keping Bi, Qingyao Ai, Yongfeng Zhang, and W. Bruce Croft. 2019. Conversational Product Search Based on Negative Feedback. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19)*. 359–368.

[4] Craig Boutilier, Kevin Regan, and Paolo Viappiani. 2009. Online feature elicitation in interactive optimization. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. 73–80.

[5] Craig Boutilier, Kevin Regan, and Paolo Viappiani. 2010. Simultaneous elicitation of preference features and utility. In *Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI '10)*.

[6] Mark J. Carman, Mark Baillie, and Fabio Crestani. 2008. Tag Data and Personalized Information Retrieval. In *Proceedings of the 2008 ACM Workshop on Search in Social Media (SSM '08)*. 27–34.

[7] Li Chen and Pearl Pu. 2006. Evaluating Critiquing-based Recommender Agents. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI '06)*. 157–162.

[8] Li Chen and Pearl Pu. 2012. Critiquing-based Recommenders: Survey and Emerging Trends. *User Modeling and User-Adapted Interaction* 22, 1-2 (April 2012), 125–150.

[9] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. 815–824.

[10] J. Shane Culpepper, Fernando Diaz, and Mark D. Smucker. 2018. Research Frontiers in Information Retrieval: Report from the Third Strategic Workshop on Information Retrieval in Lorne (SWIRL 2018). *SIGIR Forum* 52, 1 (Aug. 2018), 34–90.

[11] Leo A. Goodman and William H. Kruskal. 1954. Measures of Association for Cross Classifications. *J. Amer. Statist. Assoc.* 49, 268 (1954), 732–764.

[12] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19:1–19:19 pages.

[13] Nitin Jindal and Bing Liu. 2006. Mining Comparative Sentences and Relations. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI '06)*. 1331–1336.

[14] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*. 133–142.

[15] Wiltrud Kessler and Jonas Kuhn. 2013. Detection of Product Comparisons - How Far Does an Out-of-the-Box Semantic Role Labeling System Take You?. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP '13)*. 1892–1897.

[16] Wiltrud Kessler and Jonas Kuhn. 2014. A Corpus of Comparisons in Product Reviews. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC '14)*. 2242–2248.

[17] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (Aug. 2009), 30–37.

[18] Takeshi Kurashima, Katsuji Bessho, Hiroyuki Toda, Toshio Uchiyama, and Ryoji Kataoka. 2008. Ranking Entities Using Comparative Relations. In *Proceedings of the 19th International Conference on Database and Expert Systems Applications (DEXA '08)*. 124–133.

[19] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. 304–312.

[20] Hanze Li, Scott Sanner, Kai Luo, and Ga Wu. 2020. A Ranking Optimization Approach to Latent Linear Critiquing for Conversational Recommender Systems. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20)*. 13–22.

[21] Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS '18)*. 9748–9758.

[22] Si Li, Zheng-Jun Zha, Zhaoyan Ming, Meng Wang, Tat-Seng Chua, Jun Guo, and Weiran Xu. 2011. Product Comparison Using Comparative Relations. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '11)*. 1151–1152.

[23] Greg Linden, Steve Hanks, and Neal Lesh. 1997. Interactive Assessment of User Preference Models: The Automated Travel Assistant. In *Proceedings of the Sixth International Conference on User Modeling*. 67–78.

[24] Kai Luo, Scott Sanner, Ga Wu, Hanze Li, and Hojin Yang. 2020. Latent Linear Critiquing for Conversational Recommender Systems. In *Proceedings of The Web Conference 2020 (WWW '20)*. 2535–2541.

[25] Kai Luo, Hojin Yang, Ga Wu, and Scott Sanner. 2020. Deep Critiquing for VAE-Based Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 1269–1278.

[26] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*. 43–52.

[27] Kevin McCarthy, Yasser Salem, and Barry Smyth. 2010. Experience-Based Critiquing: Reusing Critiquing Experiences to Improve Conversational Recommendation. In *Proceedings of the 18th International Conference on Case-Based Reasoning Research and Development (ICCBR '10)*. 480–494.

[28] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fined-grained aspects. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP '19)*.

[29] Filip Radlinski, Krisztian Balog, Bill Byrne, and Karthik Krishnamoorthi. 2019. Coached Conversational Preference Elicitation: A Case Study in Understanding Movie Preferences. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL '19)*. 353–360.

[30] Filip Radlinski and Nick Craswell. 2017. A Theoretical Framework for Conversational Search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval (CHIIR '17)*. 117–126.

[31] Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM '10)*. 81–90.

[32] Anna Sepliarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. 2018. Preference elicitation as an optimization problem. In *Proceedings of the International Conference on Recommender Systems (RecSys '18)*. 172–180.

[33] Börkur Sigurbjörnsson and Roelof van Zwol. 2008. Flickr Tag Recommendation Based on Collective Knowledge. In *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*. 327–336.

[34] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. 235–244.

[35] Maksim Tkachenko and Hady W. Lauw. 2014. Generative Modeling of Entity Comparisons in Text. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM '14)*. 859–868.

[36] Daniel Tunkelang. 2009. Faceted search. *Synthesis lectures on information concepts, retrieval, and services* 1, 1 (2009), 1–80.

[37] Kasturi Dewi Varathan, Anastasia Giachanou, and Fabio Crestani. 2017. Comparative Opinion Mining: A Review. *J. Assoc. Inf. Sci. Technol.* 68, 4 (April 2017), 811–829.

[38] Jesse Vig, Shilad Sen, and John Riedl. 2012. The Tag Genome: Encoding Community Knowledge to Support Novel Interaction. *ACM Trans. Interact. Intell. Syst.* 2, 3, Article 13 (Sept. 2012), 44 pages.

[39] Jesse Vig, Matthew Soukup, Shilad Sen, and John Riedl. 2010. Tag expression: Tagging with feeling. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '10)*. 323–332.

[40] Mengting Wan and Julian J. McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. 86–94.

[41] Jason Williams, Antoine Raux, and Matthew Henderson. 2016. The Dialog State Tracking Challenge Series: A Review. *Dialogue & Discourse* 7, 3 (2016), 4–33.

[42] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep Language-Based Critiquing for Recommender Systems. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. 137–145.

[43] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep Language-Based Critiquing for Recommender Systems. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. 137–145.

[44] Yury Zemlyanskiy, Sudeep Gandhe, Ruining He, Bhargav Kanagal, Anirudh Ravula, Juraj Gottweis, Fei Sha, and Ilya Eckstein. 2021. DOCENT: Learning Self-Supervised Entity Representations from Large Document Collections. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL '21)*. 2540–2549.

[45] Shuo Zhang and Krisztian Balog. 2017. Design Patterns for Fusion-Based Object Retrieval. In *Proceedings of the 39th European conference on Advances in Information Retrieval (ECIR '17)*. 684–690.

[46] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards Conversational Search and Recommendation: System Ask, User Respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. 177–186.

[47] 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), 20:1–20:22 pages.