A Text-Based Recommender System that Leverages Explicit Affective State Preferences

Tonmoy Hasan and **Razvan Bunescu**
Department of Computer Science
University of North Carolina at Charlotte
thasan1,rbunescu@charlotte.edu

Abstract

The affective attitude of liking a recommended item reflects just one category in a wide spectrum of affective phenomena that also includes emotions such as entranced or intrigued, moods such as cheerful or buoyant, as well as more fine-grained affective states, such as "pleasantly surprised by the conclusion". In this paper, we introduce a novel recommendation task that can leverage a virtually unbounded range of affective states sought explicitly by the user in order to identify items that, upon consumption, are likely to induce those affective states. Correspondingly, we create a large dataset of user preferences containing expressions of fine-grained affective states that are mined from book reviews, and propose ACRec, a Transformer-based architecture that leverages such affective expressions as input. We then use the resulting dataset of affective states preferences, together with the linked users and their histories of book readings, ratings, and reviews, to train and evaluate multiple recommendation models on the task of matching recommended items with affective preferences. Experimental comparisons with a range of state-of-the-art baselines demonstrate ACRec's superior ability to leverage explicit affective preferences.

1 Introduction and Motivation

Traditional recommender systems (RS) leverage user preferences that are implicit in user-item rating histories in order to personalize rankings of the items presented to the user (Resnick and Varian, 1997; Park et al., 2012). As long as sufficient user-item data is available, this recommendation approach is convenient for the user as it requires little to no interaction other than providing rating feedback. This minimal interaction approach, however, can be slow to track changes in user preferences and imprecise for users with diverse preferences. Conversational recommender systems (CRS) (Thompson et al., 2004) can elicit more precise preferences by engaging in a natural dialogue with the user, an approach that more recently takes advantage of fine-tuned or appropriately prompted LLMs (Zhao et al., 2024; Lian et al., 2024). Besides their ability to carry goal driven conversations, large scale pre-trained language models also bring the advantage of having an extensive implicit memory of items and their descriptions (Penha and Hauff, 2020), which benefits content-based recommendations, as well as knowledge of correlations between item consumption histories and user preferences, which in theory could provide useful collaborative-filtering signals (He et al., 2023). To support the training and evaluation of CRS models, a number of conversational recommendation datasets have been created using either crowd-sourcing or scrapping of real interactions on dedicated internet forums. In the movie domain, for example, datasets such as INSPIRED (Hayati et al., 2020) and ReDIAL (Li et al., 2018) use a crowd-sourcing approach where workers play the roles of seekers and recommenders, whereas the Reddit-Movie (He et al., 2023) dataset contains naturally occurring dialogues where Reddit users seek and offer recommendations in the real world. Table 1 shows in narrative form prototypical examples of user preferences that were elicited in conversational recommendation dialogues. These examples illustrate a predominant aspect of CRS datasets: with very few exceptions, the elicited preferences refer to objective features of an item (colored in *teal*) or to items the user has consumed in the past. For example, users express a like or dislike attitude towards movie genres (comedies, fantasy, sci-fi, or thriller), movie directors (Kubrik), source material (DC Comics), plot and content (romantic vibe or inventive). We call these item features *objective* as they are largely independent of the user, e.g., given an arbitrary movie, most users would agree on whether the movie is a thriller, and similarly the identity of the movie director should

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 32840–32854
November 4-9, 2025

▶ I love *superhero* movies. Last night I saw Shazam and I loved the *comedy* part of it too. I usually don't like *DC movies* because they lack *humor*, but this movie had that. ▶ Can you recommend a movie like A Clockwork Orange? I liked that it was a *Kubrick movie* and that it was inventive. I didn't like that *the "future" portrayed was a little dated*.
▶ I'm into *Fantasy*. Especially *high fantasy*. I've seen Highlander. I've seen them all actually. I enjoyed them. The Matrix was great! But I didn't care for The Matrix Revolutions. It *drifted from the original story* in my opinion. Blade Runner is definitely my kind of movie. It's the perfect combo of *scifi and thriller*.
▶ Any movies like before Sunrise trilogy? I'm looking for movies where *a couple or strangers explores the ideas and world around them* and also has *a romantic vibe* to it. ▶ I've always enjoyed movies that have *dark themes* - movies that *make you uncomfortable/squirm* be it from a psychological standpoint or from just sheer brutality. Obviously with a *compelling story* or *thought provoking ideas*.

Table 1: Examples of preferences from INSPIRED (top), ReDIAL (middle), and Reddit-Movie (bottom). Objective features are shown colored in *teal*.

▶ I want a book that *makes me smile at times*, but that also *brakes my heart*. The main character is a child narrator. The book will *make me think about it for a long time*. ▶ I am looking for a young adult horror book that doesn't hold back. It is 100% shocking, scary, and over the top. I would like to *feel thrilled, disturbed, and completely entranced* while reading. The book has sympathetic characters in horrible situations. *I will not want to put the book down except to pace around the room to shake off the chills.* ▶ I am looking for a book *unlike any I've ever experienced before*. The book's *emotional impact is so heavy that, even though I'm super into it, I will have to stop for a couple of days before finishing it.* ▶ I am drawn to books about complex social issues especially those that impact families. Characters that are real, who are developed well enough so that *I can feel their emotions* and a strong story line make a book exceptional for me. It is a book that *I will not want to put down* and which evokes *a depth of emotions that takes me by surprise.*

Table 2: Examples of affective preferences generated from Goodreads reviews. Expressions of affective states are colored in *violet*, affective-cognitive states in *blue*.

be uncontroversial. The *subjective* aspect of the elicited preferences is then due solely to the user's attitude of liking or disliking particular items or their objective features.

The attitude of liking an item is the only type of affective state that is explicitly targeted by sentiment analysis or recommender systems. However, following Scherer's typology of affective states (Scherer, 2005), there are not one, but three major types of affective states that an item can impress on a user: *attitudes*, *moods*, and *emotions*. Attitudes are relatively enduring, affectively colored beliefs and dispositions towards items. The affective states induced by a salient attitude are generally weak in intensity and can be labeled with terms such as liking, disliking, loving, hating, valuing, or desiring. Moods are diffuse, low intensity affect states, characterized by a relative enduring predominance of certain types of subjective feelings. Examples are being cheerful, gloomy, listless, depressed, or buoyant. Emotions are comparatively shorter in duration but higher in intensity, and can have a strong behavioral impact, often altering ongoing behavior sequences and triggering the generation of new goals and plans. Examples include *utilitarian* emotions, such as anger, sadness, joy, fear, pride, guilt, or shame, and *aesthetic* emotions, such as wonder, awe, admiration, fascination, bliss, ecstasy, harmony, rapture, or solemnity. They can generally be elicited by a wide array of both external and internal stimulus events, such as natural phenomena, the behavior of others or one's own behavior, and sudden neuroendocrine or physiological changes. Especially relevant to this work, emotions can also be elicited by memories or images that might come to one's mind, such as the ones evoked by reading a book. These imagined representations of events are often sufficient on their own to generate strong emotions (Goldie, 2004). Table 2 shows examples of *affective* states (colored in *violet*) that were induced by reading a book, sampled from reviews in the Goodreads dataset (Wan and McAuley, 2018a). Also shown are examples of more complex *cognitive* states (colored in *blue*), named as such because they can be seen as mixing a strong affective component with other cognitive aspects, such as reasoning, attention, memory, or decision making. Note that affective states are much scarcer in Table 1, showing up only in a sample from Reddit-Movie. Compared to the other two datasets, which were created from people pretending to seek movie recommendations, Reddit-Movie contains real world, genuine recommendation requests. As such, we believe that recommender systems would provide significantly higher user satisfaction if they were able to *elicit* and *match* a wider range of desired affective states. Correspondingly, the main contribution of this paper is a novel recommender system framework where users can express a wide range of affective state preferences.

The structure of the paper is as follows: in Section 2 we introduce the task of affective-cognitive recommendations; in Section 3 we present the architecture of the corresponding recommender system; in Section 4 we introduce the book recommendation dataset used for training and evaluations,

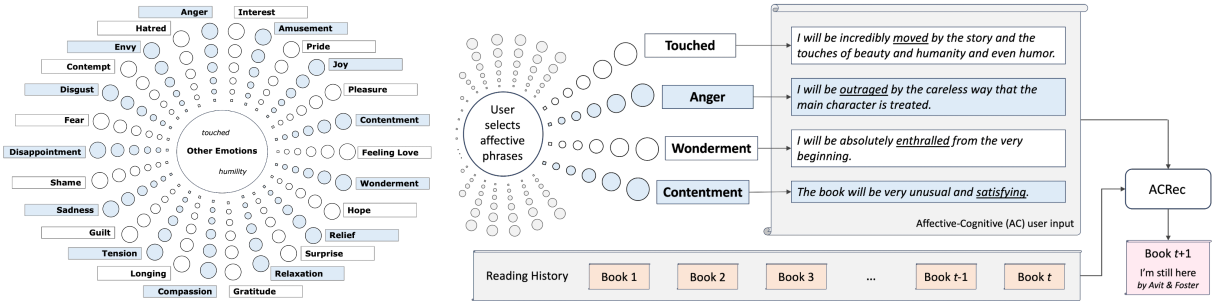

Figure 1: Selection of AC statements, starting from top emotion categories organized around the Emotion Wheel. Left: The proposed emotion wheel adapted from the Geneva Emotion Wheel. Right: The user selects emotion categories, where for each category they have the option of selecting fine-grained AC statements.

and the procedure used for generating affective-cognitive statements from reviews; in Section 5 we present experimental evaluations of the proposed architecture and a discussion of the results; in Section 6 we discuss related work. The paper ends with thoughts on limitations and conclusion.

2 The Affective-Cognitive Recommender

An Affective-Cognitive Recommender (ACRec) system takes as input data about a user, such as their history of readings, ratings, and reviews, as well as a description of the affective-cognitive (AC) states that they seek to experience upon consuming an item. Based on this input, ACRec computes a ranked list of items, where items at the top are expected to induce the AC states desired by the user, as expressed in the AC description part of the input. To input the AC description, the user can use the two modes below, separately or in combination:

1. **Generation**: the user generates the AC description as free form text (Table 2).

2. **Selection**: the user selects one or more expressions of affective states from a predefined repository of AC statements (Figure 1).

The four AC descriptions shown in Table 2 belong to the Generation mode. While useful for specifying objective features, as shown in Table 1, for most users the generation mode can be much less conducive to eliciting expressions of affective states, due to the difficulty of expressing fine-grained emotions that one would like to experience from a book they have yet to read. As such, we also propose a Selection mode that enables the user to select expressions of desired affective states from a predefined, rich repository of AC statements. To this end, we have created a large dataset of expressions of fine-grained affective states that are mined from book reviews, as detailed in Section 4. Given the thousands of AC statements contained in this dataset, for the selection process to be effective it is important that the AC statements are organized in an ontology that users can navigate easily. As such, we propose that the repository of AC statements is organized at the top level around a wheel of emotion categories, as shown at the left of Figure 1. We created this Emotion Wheel by augmenting the Geneva Emotion Wheel (GEW) (Sacharin et al., 2012) with 6 additional emotion categories that were originally mentioned in (Scherer, 2005) and that were observed to be expressed in user reviews: Contentment, Gratitude, Hatred, Hope, Relaxation (Serenity), and Tension (Stress). GEW is a theoretically derived and empirically tested instrument that was designed to measure emotional reactions to objects, events, and situations. The original 20 discrete emotion categories (or families) are organized around a circle, where the horizontal dimension indicates the emotion valence (from negative to the left to positive to the right) and the vertical dimension indicates control (from low at the bottom to high at the top). Additionally, spokes in the wheel correspond to different levels of intensity for each emotion family from low intensity (towards the center) to high intensity (toward the circumference), where each intensity level can be associated with typical words that are used to express emotions at that level. The center of the wheel is a catch all category for emotions that do not belong to any of the main categories.

Figure 1 shows an example of utilizing ACRec in the Selection mode, where the user starts by selecting 4 emotions categories. For each category they have the option of selecting an emotion word corresponding to one of the intensity levels on the spoke. The system would then present them with the set

of AC statements in the repository that correspond to that emotion category (and word if selected), possibly subcategorized according to the source of emotion, e.g. characters, writing style, or events in the book. The AC statements selected by the user using the process above form the AC description.

Together with the user history of readings, reviews, and ratings, the AC description is then provided as input to the ACRec recommendation model, which is trained to compute a ranking of all the books with respect to how likely they are to trigger in the user the affective states contained in the AC description. In the following section we introduce the architecture of an AC recommendation model that is specifically designed to leverage the AC descriptions elicited from the user.

3 The AC Recommendation Model

Given the inherent subjective nature of affective states, the recommender system needs a good model of user preferences and expectations. These will be inferred from the user data, which will consist of the history of consumed items, ratings, and reviews (notation summarized in Table 3). Let $\mathcal{U}$ be the set of users and $\mathcal{B}$ the set of items, alternatively referred to as books. For a user $u \in \mathcal{U}$, let $\mathcal{B}^u = \{b_1^u, b_2^u, \ldots, b_{t-1}^u, b_t^u\} \subseteq \mathcal{B}$ be a sequence of items in chronological order that the user u has consumed, rated, and written reviews for, up to time step t. Each item in $\mathcal{B}$ is associated with an original description and another review-based extended description. Given data about a user u up to and including the current time step t, an AC description $\mathbf{ac}$, and an arbitrary book b, the task of the ACRec system is to compute a recommendation score $y_b^u(\mathbf{ac}, b)$ that should reflect how well the book b aligns with the AC description $\mathbf{ac}$ provided by the user u.

Notations	Explanations
$\mathcal{U}, \mathcal{B}$	User and item set
$\mathcal{B}^u$	Set of items that user u has read, rated and written reviews for
$\mathbf{d}_i^u \in \mathbb{R}^d$	Embedding of i-th item's original description
$\mathbf{e}_i^u \in \mathbb{R}^d$	Embedding of i-th item's extended description
$\mathbf{r}_i^u \in \mathbb{R}^d$	Review embedding of i-th item
$\mathbf{g}_i^u \in \mathbb{R}^n$	Rating embedding of i-th item
$\mathbf{c}_i^u \in \mathbb{R}^l$	Concatenated embedding of i-th time step
$\mathbf{sp}_t^u \in \mathbb{R}^l$	User u's short-term preference at time step t
$\mathbf{lp}_t^u \in \mathbb{R}^l$	User u's long-term preference at time step t
y_b^u	Recommendation score for candidate book b for user u

Table 3: Key notation and descriptions.

The overall architecture of the proposed ACRec model is shown in Figure 2. The recommendation score $y_b^u(\mathbf{ac}, b)$ is calculated by a fully connected network with two hidden layers and one output linear layer, as shown in Equation 1, where f is the ramp function.

$$\mathbf{w}_3 f(\mathbf{W}_2 f(\mathbf{W}_1[\mathbf{lp}_t^u; \mathbf{sp}_t^u; \mathbf{s}_b; \mathbf{ac}; cos(\mathbf{ac}, \mathbf{d}_b)])) \quad (1)$$

The network takes as input representations of long-term user preferences $\mathbf{lp}_t^u$, short-term user preferences $\mathbf{sp}_t^u$, the AC description $\mathbf{ac}$, and the representation $\mathbf{s}_b$ of an arbitrary book b. The input also contains cosine similarity $cos(\mathbf{ac}, \mathbf{d}_b)$ between the AC description and the book description.

Each book is associated two textual descriptions: the *original* description from the Goodreads dataset, and an *extended* description created from the concatenation of user reviews for that book. It is important to note that reviews from training users and testing users are never used as part of extended descriptions. A book that the user u read, reviewed, and rated at time step t is then represented as the concatenation of 4 embeddings $\mathbf{c}_t^u = [\mathbf{d}_t^u; \mathbf{e}_t^u; \mathbf{r}_t^u; \mathbf{g}_t^u]$, where $\mathbf{d}_t^u$ is an embedding of the original book description, $\mathbf{e}_t^u$ is an embedding of the extended book description, $\mathbf{r}_t^u$ is an embedding of the user's review of the book, and $\mathbf{g}_t^u$ is an embedding of the rating (grade) that the user gave to the book. We first use Jina Embeddings v3 (Sturua et al., 2024), a text embedding model that produces 1024-dimensional embeddings of text containing up to 8192 tokens, with the flexibility to reduce dimensionality without compromising performance. To obtain $\mathbf{d}_t^u$, $\mathbf{e}_t^u$, and $\mathbf{r}_t^u$, we first generate their 768-dimensional Jina embeddings and reduce them to 41-dimensional embeddings using learned projection matrices W_d, W_e, and W_r, respectively. Size 5 embedding are also learned for each of the possible 5 rating values. When concatenated together, the 4 embeddings in $\mathbf{c}_t^u$ add up to a size of $3 \cdot 41 + 5 = 128$, to match the tuned hidden dimension used in the Transformer block. Overall, the user data up to the current time step t can then represented as the sequence $\langle \mathbf{c}_1^u, \mathbf{c}_2^u, ..., \mathbf{c}_{t-1}^u, \mathbf{c}_t^u \rangle$.

Computing the recommendation score for a book $b \in \mathcal{B} - \mathcal{B}_t^u$ that the user has not read yet requires a representation of the book $\mathbf{s}_b$ as well an embedding $\mathbf{ac}$ of the AC description provided by the user. The book representation $\mathbf{s}_b$ is a concatenation of its original description embedding, its extended

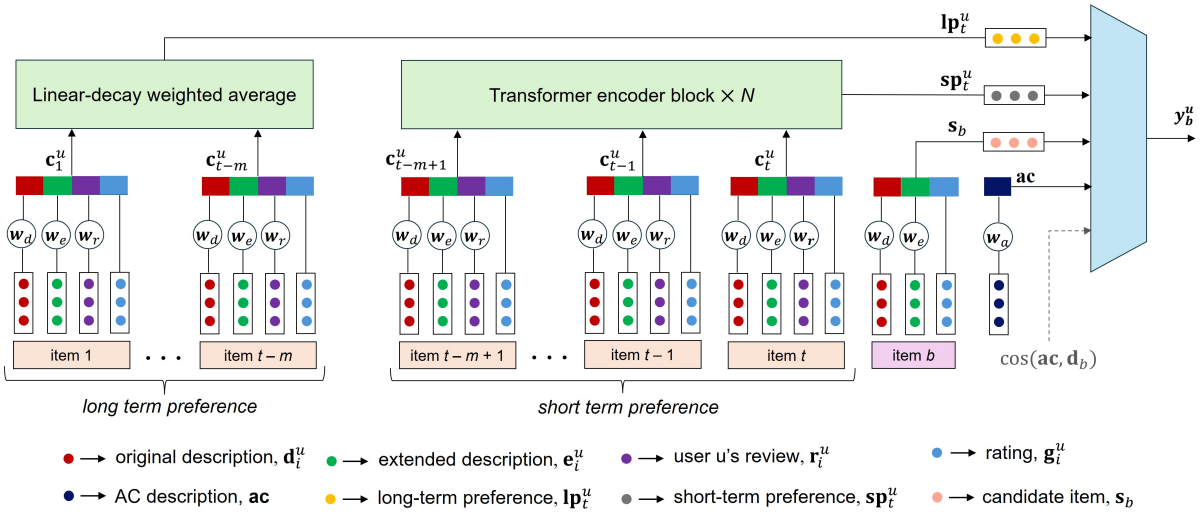

Figure 2: Overall architecture of the proposed ACRec model.

description embedding, and the embedding of the maximum rating value of 5. These embeddings are calculated using the same procedure that was used for computing the embeddings in the item representations $\mathbf{c}_t^u$. Note that we do not use a book review embedding in $\mathbf{s}_b$, as the candidate book b has not been read by the user. The AC description is embedded into a vector $\mathbf{ac}$ by projecting its Jina embedding using a separate projection matrix W_a.

3.1 User Preference Modeling

We model short-term and long-term preferences by partitioning the user's time series data in two parts:

1. **Short term preferences** $\mathbf{sp}_t^u$ are modeled by running a Transformer over the last m items $\langle \mathbf{c}_t^u, \mathbf{c}_{t-1}^u, \ldots, \mathbf{c}_{t-m+1}^u \rangle$ in this reversed order.

2. **Long term preferences** $\mathbf{lp}_t^u$ are created from a linearly weighted average of the remaining $t - m$ items $\langle \mathbf{c}_1^u, \mathbf{c}_2^u, \ldots, \mathbf{c}_{t-m}^u \rangle$.

The short-term preference embedding $\mathbf{sp}_t^u$ is set to be the hidden state computed by the last Transformer block for time step t. We use a Transformer model with hidden size of 128, 4 blocks, and the standard, fixed positional embeddings (Vaswani et al., 2017).

The long-term preferences $\mathbf{lp}_t^u$ are calculated as:

$$\mathbf{lp}_t^u = \sum_{k=1}^{t-m} \alpha_k \mathbf{c}_k^u \tag{2}$$

where the linearly decaying weights are set as $\alpha_k = 2k/((t-m)(t-m+1))$. The weights sum up to 1 and ensure that more recent items are assigned larger weights compared to more distant items.

3.2 Training Objective

To train the ACRec model, we employ the Bayesian personalized raking loss (Rendle et al., 2009) shown below:

$$\mathcal{L}(\mathbf{\Theta}) = -\sum_{u \in U} \frac{1}{T_u} \sum_{t=1}^{T_u} \frac{1}{K} \sum_{k=1}^{K} \log \sigma(y_b^u - y_k^u) \tag{3}$$

where T_u is the number of training steps, K is the number of negative samples, y_b^u is the recommendation score for the positive sample, and y_k^u is the recommendation score from a negative sample. As will be explained in Section 4 below, as positive sample we use the book known to match the AC description for the user, whereas negative samples will be drawn at random at each gradient update step from books that the user has not read. The loss is minimized using the Adam Optimizer (Kingma and Ba, 2014) with a batch size of 1.

4 The AC Book Recommendation Dataset

To train and evaluate the ACRec model, we create a large dataset that map books to AC descriptions of user preferences containing expressions of fine-grained affective states mined from book reviews contained in the Goodreads dataset (Wan and McAuley, 2018b).

4.1 User Reading Histories

We extract from the Goodreads dataset book reading histories corresponding to a total of 1,000 users. In order to address a diverse range of reading histories, we randomly select 100 users who have read between [20-50] books. We then randomly select

other 100 users for each subsequent range [51-100], [101-150], and so on up to [451-500] books, resulting in a total of 1,000 users.

Given a user reading history, we identify a time step as *useful* if the user's book rating is 4 or 5, the number of tokens of the review written by the user is at least 20, and the combined token count of the original book description and the extended book description is at least 250. We skip the first 15 time steps from the user's history, considering them a burn-in set for learning the user's reading preferences. We then select a maximum of 20 useful time steps per user, spread evenly across the user's reading history. These time steps from all 1,000 training users are then split into training, validation, and testing, as described in Section 4.2.1. Overall, the reading histories of the 1,000 users in the dataset contained 249,326 times steps in total, covering 142,892 books and 11,588 useful steps.

4.2 Affective-Cognitive Statements

We used GPT-4o to extract statements of affective-cognitive (AC) preferences from book reviews, in two phases. In Phase 1, we instruct GPT-4o to process the book reviews into statements that do not contain identifiable information such as book titles, author names, character names, direct quotations, chapter or page references, or sentences expressing negative sentiment. The extracted statements are then categorized into three types: Affective (A), Cognitive (C), Affective-Cognitive (AC). An Affective statement conveys the reader's emotional or mood-based response to the book; a Cognitive statement highlights factual, structural, or stylistic aspects, focusing on objective features without reference to emotional reactions; and an Affective-Cognitive statement integrates both, expressing emotional responses tied to specific elements of the book's content or form. Each statement should be self contained and should read as if written before reading the book, by someone seeking to learn from and experience what is described in the review, without having any prior knowledge of the actual book. We manually inspected the Phase 1 outputs for 100 randomly selected reviews and observed that, likely due to the large size of the prompt, which includes both detailed guidelines and in-context examples, combined with the often lengthy nature of the book reviews, GPT-4o occasionally misclassified C phrases as A phrases. A manual evaluation of 100 extracted AC statements, based on the rubrics in Appendix A, showed that 99 adhered to the guidelines, with one exception containing a mildly negative sentiment sentence.

To improve the classification accuracy, in Phase 2 we instructed the LLM to fix its initial categorization of statement by providing it with instructions and discriminative examples from each category. Following this refinement, we manually evaluated a random sample of 100 extracted statements in terms of how precise the model was in distinguishing phrases with affective content (A + AC) from phrases with only objective content (C). We found that GPT-4o achieved an overall precision of 88%. Overall, this approach resulted in 12,522 affective (A), 1,252 affective-cognitive (AC), and 45,935 cognitive (C) statements.

Finally, to support the user Selection of AC statements described in Section 2, all statements that contain affective content (A + AC) were mapped to the 26 + 1 categories on the Emotion Wheel, using the LLM-based approach described in Appendix E.

4.2.1 Training, Validation, and Testing

For each of the 1,000 users in the dataset, we use the last (most recent) useful time step from the user's history for testing, while the second-to-last acceptable time step is used for validation. All remaining (earlier) useful time steps are used for training. During training, for each user, we sample $K = 10$ books that the user has not read to use as negative sample in the Bayesian ranking loss computation. For each training, validation, and test time steps, we utilize as input an AC description that aggregates all A, AC, and C statements that were extracted (with the approach described in Section 4.2) from the review that the user wrote for the book at that time step, whereas the book itself is used as the ground truth positive sample.

Note that the GPT-4o LLM is not part of the ACRec architecture, and it is used solely for creating the evaluation dataset.

5 Experimental Evaluation

Validation experiments reported in Appendix B indicate that best validation performance is achieved by an ACRec model featuring 30 items in the input of the Transformer block for short-term histories, two hidden layers in the recommendation score calculation block, a weighted average block for long-term histories, the importance of AC statements as input, and cosine similarity. We compare this ACRec model against three groups of sequential baselines:

Description Type	Recommender System	Hit Ratio (HR)					NDCG			
		@1	@5	@10	@50	@100	@5	@10	@50	@100
A + AC + C	SASRec	0.1	0.7	1.1	2.4	3.2	0.4	0.5	0.8	0.9
	BERT4Rec	0.1	0.1	0.2	0.8	1.4	0.1	0.1	0.2	0.3
	UniSRec	0.0	0.1	0.3	1.6	3.2	0.0	0.1	0.4	0.6
	RecFormer	0.2	1.8	2.8	5.7	8.8	1.1	1.4	2.1	2.6
	ACRec	**5.5**	**12.3**	**15.6**	**25.6**	**29.4**	**9.1**	**10.1**	**12.4**	**13.0**
A + AC	ACRec	1.1	2.5	2.9	4.9	6.7	1.8	1.9	2.3	2.6
A	ACRec	0.8	1.9	2.1	4.5	6.4	1.3	1.4	1.9	2.2

Table 4: Hit Ratio (HR in %) and Normalized Discounted Cumulative Gain (NDCG in %) metrics when ranking all 143K items in the dataset. Results shown for A + AC + C (1,000 users), A + AC (554 users), and A (533 users).

System	A + AC + C	A + AC	A
SASRec	18.6	20.0	20.3
BERT4Rec	15.1	16.2	16.3
UniSRec	11.9	11.7	11.1
RecFormer	44.2	40.0	39.6
LLaRA	20.3	19.4	19.6
LLaRA-AC	40.9	37.6	36.8
iLoRA	20.9	21.2	21.5
iLoRA-AC	40.3	42.7	42.1
ACRec	**67.6**	**43.4**	**42.7**

Table 5: Top-1 accuracy (%) when ranking 20 randomly sampled candidate items + the 1 ground truth item.

1. ID-based models SASRec (Kang and McAuley, 2018) and BERT4Rec (Sun et al., 2019), which use only item IDs.

2. Language Model-based methods UniSRec (Hou et al., 2022) and RecFormer (Li et al., 2023), which incorporate item descriptions.

3. LLM-based hybrids LLaRA (Liao et al., 2024) and iLoRA (Kong et al., 2024), which combine item titles with ID-based architectures.

Systems in groups 1 and 2 can be used at test time to rank the full item set, while systems in group 3 use item titles in prompts and thus can rank only a limited number of items (21) due to prompt size constraints. Thus, we evaluate under two settings:

- All-item ranking (excluding previously consumed items) for groups 1 and 2.

- 21-item ranking (1 ground-truth + 20 random negatives) for all models.

To the best of our knowledge, no existing recommender system directly incorporates AC statements as input. Nevertheless, we modified the prompts of models in group 3 to enable specification of AC statements; we refer to these versions as LLaRA-AC and iLoRA-AC. Attempts to incorporate AC descriptions as a special book description in the inputs of UniSRec and RecFormer led to a worsening of their performance. More details about the baselines and their modified versions are provided in Appendix C.

Table 4 presents the performance of ACRec and baseline models in the *all-item ranking* setting, where models must rank the ground-truth item among about 143,000 candidates; under such conditions, a random guess would yield a Top-1 hit rate of only 0.000007. ACRec consistently outperforms all baselines by a large margin across all ranks and metrics. When provided with the full AC description (A + AC + C), ACRec achieves a HR@10 of 15.6% and NDCG@10 of 10.1%, whereas the strongest competing baseline, RecFormer, achieves only 2.8% and 1.4%, respectively. We also evaluate ACRec using only the AC statements that have affective content (A + AC), or statements that have solely affective content (A). The resulting drop in performance is expected, as the objective content in C statements adds substantial distinguishing information. These results highlight the importance of utilizing AC descriptions for modeling users' affective-cognitive preferences in the recommendation process.

Table 5 reports Top-1 accuracy when ranking the ground-truth item among 21 candidates, a setting that enables comparison with LLM-based hybrid models such as LLaRA and iLoRA. Once again, ACRec outperforms all other models by a large margin, achieving an accuracy of 67.6% when using the full AC description (A + AC + C); RecFormer, the strongest among non-LLM baselines, reaches only 44.2%, while the enhanced LLM variants LLaRA-AC and iLoRA-AC reach 40.9% and 42.7%, respectively. Notably, incorporating AC statements into these LLM-based models (LLaRA-AC and iLoRA-AC) leads to substantial improvements, doubling their Top-1 accuracy to 40.9% and 42.7%, respectively. This confirms that affective-cognitive information is beneficial not just for ACRec, but also

Description Type	Recommender System	Hit Ratio (HR)					NDCG			
		@1	@5	@10	@50	@100	@5	@10	@50	@100
A + AC + C	ACRec (WiD All)	5.5	12.3	15.6	25.6	29.4	9.1	10.1	12.4	13.0
	ACRec (WiD Novel)	5.4	12.2	15.7	25.0	29.1	8.9	10.0	12.1	12.8
	SASRec (WiD Novel)	0.1	0.7	0.8	1.9	2.9	0.4	0.4	0.7	0.8
	BERT4Rec (WiD Novel)	0.0	0.1	0.1	0.7	1.4	0.1	0.1	0.2	0.3
	UniSRec (WiD Novel)	0.0	0.1	0.2	1.2	2.7	0.0	0.1	0.3	0.5
	RecFormer (WiD Novel)	0.2	1.7	2.5	4.9	8.1	0.9	1.2	1.7	2.1
A + AC	ACRec (WiD All)	1.1	2.5	2.9	4.9	6.7	1.8	1.9	2.3	2.6
	ACRec (OoD Novel)	1.3	2.8	3.1	5.9	6.8	2.1	2.3	2.7	2.9
	ACRec (OoD Global)	1.1	2.2	3.1	5.9	6.4	1.7	2.1	2.5	2.0

Table 6: Top: performance on 732 WiD Novel users subset (A + AC + C) vs. all 1,000 WiD users (A + AC + C). Bottom: performance on 370 OoD Novel (A + AC) and the 125 OoD Global timeline users (A + AC).

for other recommender models. When using only the A + AC descriptions, or solely A descriptions, ACRec's performance drops while still outperforming all baselines model. Overall, the results demonstrate that when users provide richer, more informative descriptions, including both how they want to feel and what kinds of content or structure they seek, ACRec is better able to align those preferences with appropriate recommendations.

Error analysis of the ACRec model revealed several reasons for the positive sample being ranked out the the top 10, such as when the AC descriptions are overly generic, e.g., *"a well written book"*. Another issue arises when AC descriptions contain proper nouns. For instance, one AC description includes the word *Australia*. Although the description of the ground-truth book also mentions *Australia*, the model tends to recommend books where *Australia* appears more frequently or prominently in their original or extended descriptions. Finally, it is important to note that the experimental results are conservative: in reality, it is likely that there are multiple books that satisfy a given AC description, other than the ground truth book that the user actually read.

5.1 OoD and Global Timeline Generalization

To evaluate Out-of-Distribution (OoD) generalization and performance in a global timeline setting (Ji et al., 2023), we create 3 additional datasets.

WiD Novel: We identify a subset of 732 users (out of the 1,000) whose test-time ground-truth book is *novel*, meaning that the test book does not appear in the training time steps of any of the 1,000 users.

OoD Novel: We create a separate set of 370 new users, where all their test books are guaranteed to be novel and have solely A + AC descriptions. Note that these users are not used for training ARec.

OoD Global: Of the 370 OoD users, we identify 125 users whose test time step is in the future of all training time steps of the 1,000 WiD users.

In Table 6 we report results across all these settings, and compare with results from the original WiD All setting. We first examine the WiD Novel users subset (732 users), where the test-time book does not appear in any training time step. The A + AC + C results on the WiD Novel users subset show that ACRec achieves nearly identical performance on these 732 users as on all 1,000 users, indicating that leakage from book overlap between training and testing is minimal, if any. Furthermore, in the harder A + AC setting where all descriptions contain affective state preferences, ACRec attains comparable accuracy on the 125 OoD Global users and the full 370 OoD Novel users, with both results closely matching the within-distribution (WiD All) results from Table 4. These findings further suggest that user-level or time-level information leakage due is very limited, if any.

6 Related Work

In (Hasan and Bunescu, 2025), we provide a comprehensive survey on modeling attitudes, emotions, and moods for personalization of affective recommender systems. The utility of affective attitudes in recommender systems has been studied by Da'u and Salim (2019), who used a neural attention model to incorporate sentiment from user reviews. Mizgajski and Morzy (2019) introduce emotion collection methods for emotion-based news recommendations. Affective signals have also been used in short film and movie recommendations via collaborative filtering and context mining (Orellana-Rodriguez et al., 2015; Zheng, 2017). Moshfeghi and Jose (2011) represent each movie through 22 emotion categories extracted from reviews, and leverage the resulting emotion embedding for im-

proving the computation of affective similarity within a collaborative-filtering framework. Beyond text, EmoWare (Tripathi et al., 2019) leverages users' non-verbal emotional cues for context-aware video recommendations. Wang and Zhao (2022) survey affective video recommenders, while Polignano et al. (2021) propose an affective coherence model to capture emotional transitions between items. Literature reviews highlight growing interest in this area across domains, including education (Salazar et al., 2021; Katarya and Verma, 2016). However, these systems rely on users' affective responses to items consumed in the *past* to recommend items they may like. In contrast, our ACRec framework enables users to specify a broad range of desired affective states that will be induced in the *future* upon consuming the recommended items.

Beyond-accuracy objectives, such as diversity, novelty, unexpectedness, and serendipity, have gained significant attention in recommender systems. Kim et al. (2019) enhanced diversity in sequential recommendations, while the system of Li and Tuzhilin (2020) identified surprising or unexpected items. How diversity, serendipity, novelty, and coverage can enhance user satisfaction and engagement in recommender systems has been discussed along accuracy in (Kaminskas and Bridge, 2016). Serendipity is explored extensively in other works, such as (Zhang et al., 2012; Iaquinta et al., 2008; Hasan and Bunescu, 2023; Fu et al., 2023; Kotkov et al., 2016). Overall, the novelty, unexpectedness, and serendipity criteria emphasized in the systems cited above can be seen as special cases in the wide range of affective-cognitive states addressed by our system, e.g. the user's AC description can stipulate *"I would like a book that surprises me"* or *"a book that is unlike any I have read so far"*.

In conversational recommender systems, Zhou et al. (2020b) infer item-level preferences from historical interactions and attribute-level preferences from dialogue. Lei et al. (2020) propose an EAR (estimation-action-reflection) framework that leverages item attributes without incorporating long-term interaction history. Knowledge graph-based methods improve recommendation accuracy by modeling users' temporal conversations (Zhou et al., 2020a; Anelli et al., 2018; Wang et al., 2022; Zou et al., 2024; Petruzzelli et al., 2024). Zhang et al. (2024) introduce empathy by detecting emotions in conversations to align recommendations with users' affective states, identifying nine primary emotions to extract local emotion-aware entities. However, the affective-cognitive dimension remains largely underexplored. While promising, such approaches are limited in their ability to capture and induce a broad range of user emotions. In contrast, our work enables users to express and select as targets fine-grained affective states that go beyond attitudes toward objective features.

7 Conclusion and Future Work

We introduced a novel text-based recommendation setting where users express affective-cognitive (AC) states they would like to experience upon reading a book. Expressions of AC states were mined from real book reviews, and then used for the training and evaluation of recommendation architecture exploiting short-term and long-term user preferences implicit in their readings, reviews, and ratings histories. Experimental evaluations show that the proposed approach is effective at identifying books that match user's expressed affective preferences and compares favorably with state-of-the-art recommender systems. Future work includes combining ID-based with text-based information in the sequential modeling of reading histories, building an easy to navigate ontology of AC statements complete with a graphical user interface, and developing AC datasets for other domains, such as movie recommendations.

The code, datasets, and annotation guidelines are made publicly available at this repository: https://github.com/Ton-moy/ACRec.

Acknowledgments

We would like to thank the anonymous reviewers for their suggestions and constructive feedback. We thank Khushi Patel for her help with crafting part of the prompt for extracting AC statements from reviews. We are grateful to Srijan Das for providing timely access to GPT-4 models through the NAIRR Pilot initiative from the NSF (NAIRR240338). This research project has also benefited from the Microsoft Accelerating Foundation Models Research (AFMR) grant program.

Limitations

Although expressions of affective-cognitive preferences can be utilized in other domains such as movies, in this paper we only investigated their utility through the lens of book recommendations. Furthermore, the experimental results were obtained

on users having a sufficiently large reading history to enable learning of short-term and long-term user preferences. As such, performance for all tested systems is likely to be lower when text data is limited or unavailable, such as in cold-start scenarios. As such, in future work we plan to investigate alternative ways of modeling users' affective states when the user history is limited. We also plan to explore the applicability of affective-cognitive descriptions to the movie recommendation domain.

Ethical Considerations

The AC dataset presented in this paper has been derived from a publicly available dataset that has been utilized in previous studies. This dataset does not contain any sensitive user information, ensuring compliance with ethical standards. Nevertheless, during the generation of affective-cognitive descriptions from original reviews, we encountered instances where the LLM did not allow processing some reviews, displaying the error: "violated content policies due to the presence of adult or hate content." Such reviews were omitted from the dataset.

References

Vito Walter Anelli, Pierpaolo Basile, Derek Bridge, Tommaso Di Noia, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Markus Zanker. 2018. Knowledge-aware and conversational recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 521–522, New York, NY, USA. Association for Computing Machinery.

Aminu Da'u and Naomie Salim. 2019. Sentiment-aware deep recommender system with neural attention networks. *IEEE Access*, 7:45472–45484.

Zhe Fu, Xi Niu, and Mary Lou Maher. 2023. Deep learning models for serendipity recommendations: A survey and new perspectives. *ACM Comput. Surv.*, 56(1).

Peter Goldie. 2004. The life of the mind: commentary on "Emotions in everyday life". *Social Science Information*, 43(4):591–598. Publisher: SAGE Publications Ltd.

Tonmoy Hasan and Razvan Bunescu. 2023. Topic-level bayesian surprise and serendipity for recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 933–939, New York, NY, USA. Association for Computing Machinery.

Tonmoy Hasan and Razvan Bunescu. 2025. A survey of affective recommender systems: Modeling attitudes, emotions, and moods for personalization. *arXiv preprint arXiv:2508.20289*.

Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyan Shi, and Zhou Yu. 2020. INSPIRED: Toward Sociable Recommendation Dialog Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8142–8152, Online. Association for Computational Linguistics.

Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian Mcauley. 2023. Large Language Models as Zero-Shot Conversational Recommenders. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, pages 720–730, New York, NY, USA. Association for Computing Machinery.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 585–593, New York, NY, USA. Association for Computing Machinery.

Leo Iaquinta, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, Michele Filannino, and Piero Molino. 2008. Introducing serendipity in a content-based recommender system. In *2008 Eighth International Conference on Hybrid Intelligent Systems*, pages 168–173.

Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2023. A critical study on data leakage in recommender system offline evaluation. *ACM Trans. Inf. Syst.*, 41(3).

Marius Kaminskas and Derek Bridge. 2016. Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Trans. Interact. Intell. Syst.*, 7(1).

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206.

Rahul Katarya and Om Prakash Verma. 2016. Recent developments in affective recommender systems. *Physica A: Statistical Mechanics and its Applications*, 461:182–190.

Yejin Kim, Kwangseob Kim, Chanyoung Park, and Hwanjo Yu. 2019. Sequential and diverse recommendation with long tail. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2740–2746. International Joint Conferences on Artificial Intelligence Organization.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Xiaoyu Kong, Jiancan Wu, An Zhang, Leheng Sheng, Hui Lin, Xiang Wang, and Xiangnan He. 2024. Customizing language models with instance-wise lora for sequential recommendation. In *Advances in Neural Information Processing Systems*, volume 37, pages 113072–113095. Curran Associates, Inc.

Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. 2016. A survey of serendipity in recommender systems. *Knowledge-Based Systems*, 111:180–192.

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, page 304–312, New York, NY, USA. Association for Computing Machinery.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 1258–1267, New York, NY, USA. Association for Computing Machinery.

Pan Li and Alexander Tuzhilin. 2020. Latent unexpected recommendations. *ACM Trans. Intell. Syst. Technol.*, 11(6).

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Jianxun Lian, Yuxuan Lei, Xu Huang, Jing Yao, Wei Xu, and Xing Xie. 2024. RecAI: Leveraging Large Language Models for Next-Generation Recommender Systems. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, pages 1031–1034, New York, NY, USA. Association for Computing Machinery.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-recommendation assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 1785–1795, New York, NY, USA. Association for Computing Machinery.

Jan Mizgajski and Mikołaj Morzy. 2019. Affective recommender systems in online news industry: how emotions influence reading choices. *User Modeling and User-Adapted Interaction*, 29(2):345–379.

Yashar Moshfeghi and Joemon M. Jose. 2011. Role of emotional features in collaborative recommendation. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval - Volume 6611*, ECIR 2011, page 738–742, Berlin, Heidelberg. Springer-Verlag.

Claudia Orellana-Rodriguez, Ernesto Diaz-Aviles, and Wolfgang Nejdl. 2015. Mining affective context in short films for emotion-aware recommendation. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, HT '15, page 185–194, New York, NY, USA. Association for Computing Machinery.

Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. 2012. A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11):10059–10072.

Gustavo Penha and Claudia Hauff. 2020. What does BERT know about books, movies and music? Probing BERT for Conversational Recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, pages 388–397, New York, NY, USA. Association for Computing Machinery.

Alessandro Petruzzelli, Alessandro Francesco Maria Martina, Giuseppe Spillo, Cataldo Musto, Marco De Gemmis, Pasquale Lops, and Giovanni Semeraro. 2024. Improving transformer-based sequential conversational recommendations through knowledge graph embeddings. In *Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '24, page 172–182, New York, NY, USA. Association for Computing Machinery.

Marco Polignano, Fedelucio Narducci, Marco de Gemmis, and Giovanni Semeraro. 2021. Towards emotion-aware recommender systems: an affective coherence model based on emotion-driven behaviors. *Expert Systems with Applications*, 170:114382.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, page 452–461, Arlington, Virginia, USA. AUAI Press.

Paul Resnick and Hal R. Varian. 1997. Recommender systems. *Communications of the ACM*, 40(3):56–58.

V. Sacharin, Schlegel K., and Scherer K. R. 2012. Geneva Emotion Wheel rating study. Unpublished report. University of Geneva: Swiss Center for Affective Studies.

Camilo Salazar, Jose Aguilar, Julián Monsalve-Pulido, and Edwin Montoya. 2021. Affective recommender systems in the educational field. a systematic literature review. *Comput. Sci. Rev.*, 40(C).

Klaus R. Scherer. 2005. What are emotions? And how can they be measured? *Social Science Information*, 44(4):695–729. Publisher: SAGE Publications Ltd.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, et al. 2024. jina-embeddings-v3: Multilingual embeddings with task lora. *arXiv preprint arXiv:2409.10173*.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 1441–1450, New York, NY, USA. Association for Computing Machinery.

Cynthia A. Thompson, Mehmet H. Göker, and Pat Langley. 2004. A personalized system for conversational recommendations. *J. Artif. Int. Res.*, 21(1):393–428.

Abhishek Tripathi, T. S. Ashwin, and Ram Mohana Reddy Guddeti. 2019. Emoware: A context-aware framework for personalized video recommendation using affective video sequences. *IEEE Access*, 7:51185–51200.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Mengting Wan and Julian McAuley. 2018a. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 86–94, New York, NY, USA. Association for Computing Machinery.

Mengting Wan and Julian McAuley. 2018b. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 86–94, New York, NY, USA. Association for Computing Machinery.

Dandan Wang and Xiaoming Zhao. 2022. Affective video recommender systems: A survey. *Frontiers in Neuroscience*, 16.

Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 1929–1937, New York, NY, USA. Association for Computing Machinery.

Xiaoyu Zhang, Ruobing Xie, Yougang Lyu, Xin Xin, Pengjie Ren, Mingfei Liang, Bo Zhang, Zhanhui Kang, Maarten de Rijke, and Zhaochun Ren. 2024. Towards empathetic conversational recommender systems. *arXiv preprint arXiv:2409.10527*.

Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. 2012. Auralist: introducing serendipity into music recommendation. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, page 13–22, New York, NY, USA. Association for Computing Machinery.

Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2024. Recommender Systems in the Era of Large Language Models (LLMs). *IEEE Transactions on Knowledge and Data Engineering*, (01):1–20. Publisher: IEEE Computer Society.

Yong Zheng. 2017. Affective prediction by collaborative chains in movie recommendation. In *Proceedings of the International Conference on Web Intelligence*, WI '17, page 815–822, New York, NY, USA. Association for Computing Machinery.

Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020a. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1006–1014, New York, NY, USA. Association for Computing Machinery.

Kun Zhou, Wayne Xin Zhao, Hui Wang, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020b. Leveraging historical interaction data for improving conversational recommender system. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 2349–2352, New York, NY, USA. Association for Computing Machinery.

Jie Zou, Aixin Sun, Cheng Long, and Evangelos Kanoulas. 2024. Knowledge-enhanced conversational recommendation via transformer-based sequential modelling. *ACM Trans. Inf. Syst.*

A Generation of AC statements

Due to size constraints, the prompts and the few-shot examples used with the GPT-4o model in order to generate AC statements are made available at this link[1]. The following criteria were applied to manually evaluate the quality of the GPT-4o-generated AC statements:

- Does the generated AC text exclude identifiable book information, e.g. the title?
- Does the generated AC text exclude identifiable author information, e.g. the author's name?
- Does the generated AC text exclude identifiable character information, e.g. the character's name?

[1] https://github.com/Ton-moy/ACRec

- Does the generated AC text omit sentences directly taken from the book?
- Does the generated AC text exclude sentences with negative or partially negative connotation towards the book?
- Does the generated AC text exclude chapter numbers, chapter names, and page number?

B Validation Experiments

We used the validation time steps to evaluate 4 models: **Cosine**, **FCN**, **ACRec**, and **ACRecCos**. To evaluate each model's performance, we use HitRate@10 and NDCG@10: HR@10 is the fraction of times the ground-truth positive book appears among top-10 recommended books, whereas NDCG@10 is a position aware metric calculated as the inverse of $\log_2(r_b + 1)$, where r_b is the rank of the ground-truth positive book b if among the top-10 items, otherwise 0.

Model	HR@10	NDCG@10
Cosine	**81.6**	**66.2**
FCN-0	39.0	25.1
FCN-1	67.6	42.7
FCN-2	75.8	49.1
FCN-3	**77.2**	**52.1**
FCN-4	75.6	51.1
ACRec-1	81.7	54.3
ACRec-2	**86.5**	**60.8**
ACRec-3	86.4	60.3
ACRec-2+Cos	**88.3**	**67.1**

Table 7: Validation performance over 1,000 users.

In the Cosine model, we create a combined book description by concatenating the books' original and extended descriptions. For each test sample, we calculate the cosine similarity between the Jina embedding of the AC description $\mathbf{ac}$ and the Jina embeddings of the combined book description $\mathbf{d}_b$. The 101 test samples are then ranked based on their cosine similarity score.

In the FCN model, the Jina embeddings of the original book description, extended book description, and AC description are ran through three separate linear projection layers, followed by a ramp activation function, reducing their dimensionality to 42, 43 and 43, respectively. These projected embeddings are then concatenated to form a 128-dinmensional vector, which is then passed through n hidden layers before a final linear layer produces the recommendation score. During validation we experimented with $n \in \{0, 1, 2, 3, 4\}$, with the respective models being called FCN-n. Based on the validation performance showed in Table 7, we selected FCN-3 as the final FCN model for testing.

The base ACRec model is described in Section 3. During validation we considered three variants ACRec-1, ACRec-2, and ACRec-3, that were using 1, 2, or 3 hidden layers, respectively, in the fully connected network computing the recommendation score. Based on the validation performance showed in Table 7, we selected ACRec-2 as the base ACRec model for testing. The short-term history was set to the last 30 items, based on the validation results shown in Table 8. The higher performance obtained by the ACRec-2 model in Table 7 vs. Table 8 shows the impact of utilizing long-term preferences in the model. Additional hyperparameters and their tuning procedures are described in Appendix D.

According to the validation results Table 7, Cosine achieves better NDCG than the best ACRec model. Therefore, we created a combined model ACRecCos that uses the cosine similarity as additional input to the fully connected network computing the recommendation score, as shown through the dashed line in Figure 2. Correspondingly, the recommendation score y_b^u for the ACRecCos model is then computed by Equation 1.

Given that incorporating Cosine improved the performance of ACRec, in this paper we use ACRec to refer to the recommendation model that utilizes cosine as a feature.

C Baselines

For all baseline models, we follow the hyperparameter settings, training procedures, and fine-tuning strategies as described in their respective papers. A brief overview of each baseline is provided below.

SASRec: This (Kang and McAuley, 2018) is an ID-based sequential recommender model that applies unidirectional self-attention to capture user-item interaction patterns over time. Each item is represented by a learned ID embedding combined with a positional embedding, without relying on textual features.

BERT4Rec: This (Sun et al., 2019) is an ID-based model that employs bidirectional self-attention with masked language modeling (MLM) to learn user and item embeddings without using textual content. It models user sequential behavior by randomly masking items and predicting them based on both left and right context.

		Number m of latest items for short-term history									
		15	20	25	30	35	40	45	50	55	60
ACRec-2	HR@10	79.2	81.6	80.4	**82.4**	80.0	77.9	82.0	81.3	79.5	75.6
	NDCG@10	53.9	53.4	53.9	**56.3**	52.6	51.5	55.1	55.2	52.6	49.9

Table 8: Performance of ACRec-2 on validation time steps without long-term preference.

UniSRec: This (Hou et al., 2022) is a text-based model that uses item descriptions instead of ID embeddings to learn item representations with a pre-trained language model like BERT. It applies bidirectional self-attention and contrastive pretraining to learn universal sequence representations, enabling transfer across domains using parametric whitening and a Mixture-of-Experts adaptor.

RecFormer: This (Li et al., 2023) is a text-based model that formulates each item as a "sentence" by flattening key-value attribute pairs (e.g., title, category, brand, color) into natural language input, entirely eliminating the use of item IDs. It uses a bidirectional Transformer based on Longformer to learn item and sequence representations from text, and is pre-trained with masked language modeling and item-item contrastive tasks.

LLaRA: This (Liao et al., 2024) is a hybrid model that integrates item ID-based embeddings from a traditional sequential recommender (e.g., SASRec) with item text metadata (e.g., title) using a trainable projector to align both into the input space of a pretrained LLM. It uses LLaMA 2-7B as the language model backbone and applies LoRA for parameter-efficient fine-tuning. To effectively incorporate both textual and behavioral signals, LLaRA employs curriculum prompt tuning, which begins with text-only prompts and progressively introduces behavioral tokens during training. The following is the original movie recommender prompt used by LLaRA:

- LLaRA Movie Recommender Prompt: "This user has watched [HistoryHere] in the previous. Please predict the next movie this user will watch. Choose the answer from the following 21 movie titles: [CansHere]. Answer: "

For training and evaluation of the LLaRA model in our book recommendation setting, we adapted the prompt as follows:

- LLaRA Book Recommender Prompt: "This user has read [HistoryHere] in the previous. Please predict the next book this user will read. Choose the answer from the following 21 book titles: [CansHere]. Answer: "

LLaRA-AC: To laverage users' AC statements, we modified the prompt accordingly and trained the model using the same user–item interaction steps as in the ACRec model. During training, we utilized a combination of A, C, and AC phrases and we then evaluated the model on both the A + C + AC and the A + AC combination. The prompt we used here is as follows:

- LLaRA-AC Prompt: "This user has read [HistoryHere] in the previous. Now this user wants to read a book that has the following characteristics: [ac_phrases]. Please predict the next book this user will read. Choose the answer from the following 21 book titles: [CansHere]. Answer: "

iLoRA: It (Kong et al., 2024) is an extension of LLaRA that integrates a Mixture of Experts (MoE) mechanism into the standard LoRA framework to address user behavior variability in sequential recommendation. Rather than applying a uniform LoRA module across all sequences, iLoRA dynamically assembles instance-specific LoRA modules by splitting the low-rank adaptation matrices into multiple experts. We train and evaluate iLoRA using the same procedure as LLaRA. The following is the default book recommendation prompt format used in iLoRA:

- iLoRA Book Recommender Prompt: "This user has read [HistoryHere] in the previous. Please predict the next book this user will read. The book title candidates are [CansHere]. Choose only one book from the candidates. The answer is "

iLoRA-AC: To leverage AC statements, we extend iLoRA prompt, which includes the user's affective-cognitive statements:

- iLoRA-AC Prompt: "This user has read [HistoryHere] in the previous. Now this user wants to read a book that has the following characteristics: [ac_phrases]. Please predict the next

book this user will read. The book title candidates are [CansHere]. Choose only one book from the candidates. The answer is "

D Hyperparameter Tuning, Model Size and Computing Infrastructure

We implement our proposed model utilizing the Pytorch framework in Python. We perform a grid search on validation examples for: dropout in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, learning rate in $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$, number of encoder blocks in $\{1, 2, 3, 4, 5, 6\}$, the number m of items for modeling users' short-term preferences in $\{15, 20, 25, 30, 35, 40, 45, 50, 55, 60\}$. Upon performing the grid search, the dropout is set to 0.2, the learning rate to 0.0001, the number of encoder blocks to 4, and the number of latest items to 30 for our model. The hidden dimension in the transformer encoder block is 128. We also tried hidden dimensions of 64, 256, 512, however, the model with a dimension size of 128 gave the best results on validation data. Correspondingly, the total number of parameters in the ACRec model is 551,294. The hyperparameter tuning and experimental evaluations were conducted in a high-performance computing cluster on a A40 GPU with 48 GB memory, running for approximately 40 hours.

E Mapping Affective Statements to the Emotion Wheel

To support the user Selection of AC statements described in Section 2, all statements that contain affective content (A + AC) were mapped to the 26 + 1 categories on the Emotion Wheel. For this task, we employed GPT-4o to assign each affective (A or AC) statement to one or more of 26 primary emotion categories. The prompt also instructed the model to consider not only a set of typical words for that emotion category, but also synonymous words or expressions that convey similar emotional meanings, to improve classification coverage. Any AC statement expressing a reader emotion not falling into the 26 predefined categories was labeled as "Other." If a phrase expressed multiple distinct emotions, it was assigned to all relevant categories within the taxonomy.

The full prompt used for this classification is available in the repository[2]. To assess the quality of this emotion-based categorization, we manually annotated a random sample of 100 phrases and compared the results to GPT-4o's outputs. The model achieved an accuracy of 93%. The distribution of AC statements with affective content across the 26 emotion categories is summarized in Table 9.

Emotion Category	Count
Interest	3310
Sadness	1341
Surprise	1256
Feeling love	1027
Amusement	1012
Wonderment	935
Pleasure	910
Longing	800
Compassion	744
Joy	643
Hope	620
Other	591
Tension	579
Fear	557
Contentment	312
Disappointment	165
Anger	155
Relief	120
Relaxation	107
Disgust	93
Gratitude	71
Hatred	53
Pride	35
Guilt	21
Envy	17
Contempt	17
Shame	16

Table 9: Emotion Category Distribution.

[2] https://github.com/Ton-moy/ACRec