

Singapore Management University

## Institutional Knowledge at Singapore Management University

---

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

---

12-2023

### Memory network-based interpreter of user preferences in content-aware recommender systems

Nhu Thuat TRAN

Singapore Management University, nttran.2020@phdcs.smu.edu.sg

Hady W. LAUW

Singapore Management University, hadywlaw@smu.edu.sg

Follow this and additional works at: [https://ink.library.smu.edu.sg/sis\\_research](https://ink.library.smu.edu.sg/sis_research)



Part of the [Databases and Information Systems Commons](#), [Numerical Analysis and Computation Commons](#), and the [Software Engineering Commons](#)

---

#### Citation

TRAN, Nhu Thuat and LAUW, Hady W.. Memory network-based interpreter of user preferences in content-aware recommender systems. (2023). *ACM Transactions on Intelligent Systems and Technology*. 14, (6), 1-28.

Available at: [https://ink.library.smu.edu.sg/sis\\_research/8340](https://ink.library.smu.edu.sg/sis_research/8340)

This Journal Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email [cherylds@smu.edu.sg](mailto:cherylds@smu.edu.sg).



# Memory Network-Based Interpreter of User Preferences in Content-Aware Recommender Systems

NHU-THUAT TRAN and HADY W. LAUW, School of Computing and Information Systems, Singapore Management University, Singapore

This article introduces a novel architecture for two objectives *recommendation* and *interpretability* in a unified model. We leverage textual content as a source of interpretability in content-aware recommender systems. The goal is to characterize user preferences with a set of human-understandable attributes, each is described by a single word, enabling comprehension of user interests behind item adoptions. This is achieved via a dedicated architecture, which is interpretable by design, involving two components for recommendation and interpretation. In particular, we seek an *interpreter*, which accepts holistic user's representation from a *recommender* to output a set of activated attributes describing user preferences. Besides encoding interpretability properties such as fidelity, conciseness and diversity, the proposed memory network-based *interpreter* enables the generalization of user representation by discovering relevant attributes that go beyond her adopted items' textual content. We design experiments involving both human- and functionally-grounded evaluations of interpretability. Results on four real-world datasets show that our proposed model not only discovers highly relevant attributes for interpreting user preferences, but also enjoys comparable or better recommendation accuracy than a series of baselines.

CCS Concepts: • **Information systems** → **Recommender systems; Collaborative filtering;**

Additional Key Words and Phrases: Interpretable user preferences, content-aware recommendation, memory network

## ACM Reference format:

Nhu-Thuat Tran and Hady W. Lauw. 2023. Memory Network-Based Interpreter of User Preferences in Content-Aware Recommender Systems. *ACM Trans. Intell. Syst. Technol.* 14, 6, Article 108 (November 2023), 28 pages.

<https://doi.org/10.1145/3625239>

## 1 INTRODUCTION

Recommender systems are prevalent in various domains including e-commerce, news, social media, and so on. The methodologies range from matrix factorization [25] to attention-based historical aggregation [9], autoencoder-based models [33, 53, 69], and graph-based models [18, 66]. Two issues that commonly plague recommender systems are *sparsity* and *lack of interpretability*.

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-020).

Authors' addresses: N.-T. Tran and H. W. Lauw, N.-T. Tran and H. W. Lauw, School of Computing and Information Systems, Singapore Management University, 80 Stamford Road, Singapore 178902; e-mails: [ntran.2020@phdcs.smu.edu.sg](mailto:ntran.2020@phdcs.smu.edu.sg), [hadywlawu@smu.edu.sg](mailto:hadywlawu@smu.edu.sg).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2157-6904/2023/11-ART108 \$15.00

<https://doi.org/10.1145/3625239>

The former is due to the few observations relative to the large number of users or items, resulting in difficulty in building models that are sufficiently generalizable, particularly for long-tail instances. The latter is due to the abstract nature of latent representations of users and items derived from representation learning methods.

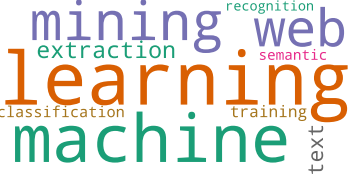
Side information such as textual content could provide another pathway for establishing similarities across users or items to improve recommendation accuracy. Representative content-aware recommender system models, including [28, 32, 42, 62, 74], mainly employ textual content to resolve sparsity. Going beyond, this work employs textual content as a source of interpretability of user's preferences. We consider two objectives, *recommendation* and *interpretability*, in a unified model, realized by two components respectively. *Recommender* focuses on learning to recommend items, while *interpreter* accepts user's representation from *recommender* as input and outputs an interpretation of user interests.

To implement our idea, we design a novel architecture, which is inspired by **Supervised Learning with Interpretation (SLI)** [48]. We realize *recommender* as an autoencoder, learning non-linear representation of user at the hidden layer and predicting user-item interactions at the output layer. In the scope of this article, we focus on two variants of autoencoder-based *recommender*, namely AutoRec [53], and CDAE [69]. Autoencoder is chosen to help reducing the learning complexity as described in Section 3.3. Our architecture design is flexible enough to plug in other neural recommenders. We verify this applicability by examining non-autoencoder recommender called *DirectAU*. Pertaining to *interpreter*, a key-value memory network [46, 56] lies at the core. It stores two matrices of the same size, namely key matrix and value matrix. *Key matrix* is a vocabulary-sized dictionary, whose each element stores the representation of a word, also called an *attribute*. The definition in [13] refers a single word as a *cognitive chunk*, i.e., unit of interpretability. Without any other specification, in this article, we term *single word*, *attribute* and *cognitive chunk* interchangeably. *Value matrix*, on the other hand, stores another representation for each word. Generally, the difference is that key matrix stores representations from textual content signals, i.e., item-word relationships, while value matrix stores representations from collaborative filtering signals. Key matrix acts a 'translator' in the sense that multiplying user representation from recommender with key matrix is equivalent to translating user representation into word space and high-similarity words captures user's preferences well. Value matrix stores building blocks to build up user representation based on generated words. The score produced by multiplying key matrix and user representation is the weight to aggregate building blocks from value matrix to output interpreter-based user vector. Key-value memory network brings two pertinent advantages. Firstly, it is flexible so that one can store n-grams as cognitive chunks. However, a larger-sized dictionary requires larger memory consumption and may slow the learning process. Secondly, by storing all words in the vocabulary, interpreter can generalize user's representation by attending to relevant words that go beyond user's adopted items' texts. We empirically demonstrate that this also benefits recommendation performance (see Section 4).

For a sense of the kind of interpretable representation we seek, Table 1 shows how given a user's historical adoptions, in this case titles of academic articles (left column), we arrive at a list of inferred natural language words (right) underlying the given user's preferences, presented as a word cloud. This is not merely keyword extraction, as some of these words may not necessarily have occurred within the adopted titles.

Our work is widely divergent from existing works in explainable recommendation and post-hoc explanation. The former concerns the underlying reasons behind a *single* user-item interaction while ours makes sense of user preferences holistically underlying their interactions with a *set of items*. Our model is *interpretable by design*, distinguishing itself from post-hoc explanation, which has been criticized for the lack of faithfulness of interpretation [52].

Table 1. Inferred Words for a user based on their Adoptions

Titles of Adopted Items	Inferred Words
<ol style="list-style-type: none"> <li>1. A Brief Survey of Web Data Extraction Tools</li> <li>2. A Tutorial on Support Vector Machines for Pattern Recognition</li> <li>3. Adaptive information extraction</li> <li>4. Automatic web news extraction using tree edit distance</li> <li>5. A Survey of Web Information Extraction Systems</li> <li>6. Relational Learning of Pattern - Match Rules for Information Extraction</li> <li>7. BoosTexter: A Boosting - based System for Text Categorization</li> <li>8. Pattern Recognition and Machine Learning (Information Science and Statistics)</li> </ol>	

**Contributions.** In this work, we make the following contributions. *Firstly*, we present a novel architecture, called INTEREC (Section 3), which stands for *Memory-based INTERpretable representation for user-oriented content-aware RECommendation*, a dedicated and unified architecture for both recommendation and interpretability. To the best of our knowledge, this is the first work that incorporates textual content-based interpreter of user preferences into a recommendation model. *Secondly*, we innovatively use key-value memory network as means of interpretation. The proposed architecture is flexible, so various neural recommenders as well as various types of attributes can be leveraged for interpretation. *Thirdly*, we investigate a technique to promote conciseness of interpretability, which also brings recommendation performance gain. *Lastly*, we empirically demonstrate a significant advancement over comparable baselines on four datasets in accuracy and interpretability, quantitatively and qualitatively (Section 4).

## 2 RELATED WORK

**Content-Aware Recommender Systems.** The line of research that incorporates item textual content into recommendation models includes CTR [60] with text modeling based on LDA [2], CDL [62] based on stacked denoising auto-encoder, ConvMF [28] based on convolutional neural networks, and CVAE [32] based on variational autoencoder. Though they vary in the text modeling, they have in common a regularization framework that encourages the text representation of an item to be close to its collaborative filtering representation. The goal of these works is to mainly resolve sparsity of user-item interaction data, leading to better recommendation performance. Subsequent works include JSR [74] that jointly predicts user-item interaction and reconstructs item textual description; GATE [42] that leverages attention network to model textual content of items and gating mechanism to combine collaborative filtering and content-based representation. These works also aim at achieving higher accuracy. Our work is distinct in a couple of ways. For one, existing works mainly employ textual content to resolve sparsity while we focus on both interpretation of user’s preferences and sparsity alleviation. For another, our model generates a personalized set of words describing user interests, which achieves higher level of interpretability, while existing works mainly employ textual content on item side. For parity, we compare against baselines in both item-oriented and user-oriented fashions.

Our work is also related to the use of heterogeneous side information to resolve data sparsity and cold-start problems, leading to better recommendation accuracy as well as improving interpretability. For instance, **knowledge graph (KG)** provides rich item attributes to characterize items and enhance user-item relationships. Notable works include path-based models [22, 67], regularization based-models [36, 76], and GNN models [63, 65]. On the other hand, social connections provide useful information to characterize user’s preferences based on their friendships on social platforms. Representative approaches include fusing user representations from social domain and item preference domain [6, 24, 70] or leveraging graph neural networks to model user-user and/or user-item connections [14, 39, 68]. Recently, thanks to the advance of learning

from **heterogeneous information network (HIN)** [72], researchers have designed novel mechanisms to model the heterogeneity of users, items and their associated information, e.g., user social connections, item relations, from a heterogeneous network [3, 4, 10]. Our model INTEREC distinguishes itself with a couple of points. For one, our motivation stems from interpretability perspective, where textual content of items is employed as the source of interpretability in our dedicated *interpreter* to discover related words capturing user preferences. Despite can be used for interpretability like textual content, KG is costly to construct and not all benchmark datasets accompany KG. For another, INTEREC is able to generate relevant textual-based interpretation of user's preferences relying solely on item textual information, which achieves lower level of model complexity than those using both item and user side information.

**Explainable Recommendation.** Recently, there are also active efforts [77] in making recommendations more explainable to consumers. The gist is in accompanying a recommendation of an item to a user with an explanation, which could be in various forms such as text [64, 78], rules [43], social graph [50], and visual imagery [37]. Our focus in this article is in interpreting the preferences of a user as a whole in terms of keywords to provide some interpretability to the workings of the content-aware recommendations. It is not our intention to explain individual item-wise recommendation instances.

**User Profiling.** There exist several works that seek to profile users for recommender systems. [15, 40] infers topics of interest to a user, both static, and dynamic. In [16], the authors profile user as a hierarchy of interactions, item level and category level. In contrast, we focus on words as units of interpretation. Outside of recommendation, user profiling is also investigated in Twitter [35], streaming short texts [34], or Question Answering [41].

**Interpretable AI.** Generally, studies on interpretable AI can be broadly categorized into two groups [38]. The first group relies on internal structure to interpret the working of a machine learning system [5, 31, 58]. The second group, *post-hoc interpretation*, including [51, 54, 75], treats machine learning model as blackbox and attempt to explain the model outputs. Our work fits into the former group of *interpretable by design*.

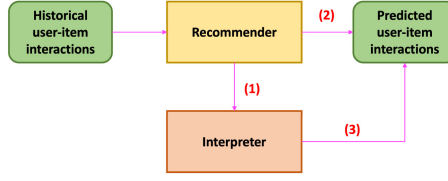
Studies on interpretability in recommender systems include [55] using CNN-based attention network to model globally and locally user's preference from reviews, [23] exploiting attention network to model the content features of movies, and [47] projecting item's representation into interpretable space to infer user preference on item's features. Our work is distinct in deriving top- $k$  words as "interpretation" for a user's latent representation.

Pertaining to dictionary of attribute-based interpretability, representative works include [12, 26, 30]. These are not comparable with ours since the dictionary of attribute is assumed to be available in advance. FLINT [48] is dedicated for multi-class image classification while ours is applied to recommendation. Hence, there is a wide difference in constructing dictionary of attributes and visualization of interpretability. Moreover, we use memory network for interpretation while FLINT employs softmax function over attribute dictionary.

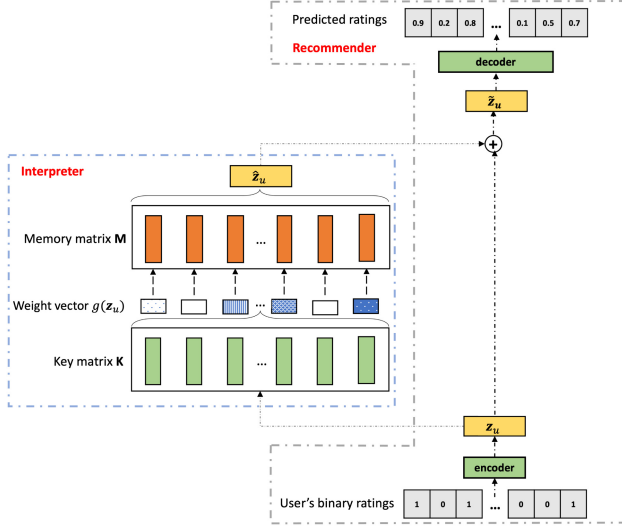
**Neural Attention-Based Recommender Systems.** ACF [9] leverages attention to model multi-media contents in collaborative filtering. [11] leverages memory network [56] to model dynamic user preferences to improve sequential recommendation. LRML [57] employs memory network to generate relation vector between user and item in metric learning so as to improve recommendation. Our novelty comes from the employment of key-value memory network [46, 56], in which attention lies at the heart, to build up *interpreter* from textual content.

### 3 METHODOLOGY

Our proposed architecture is illustrated in Figure 1 and the list of notations is presented in Table 2. The input includes binary interaction matrix  $\mathbf{R} \in \{0, 1\}^{M \times N}$ ,  $M, N$  are the number of users and



(a) Overview of INTEREC. (1) We design an *interpreter*, which accepts user representation from *recommender*, to interpret user's preferences. Output of *interpreter* (3) and that of *recommender* (2) is combined to further improve recommendation performance.



(b) A realization of INTEREC. *Recommender* is an autoencoder while *interpreter* is a memory network including two matrices  $\mathbf{K}$  and  $\mathbf{M}$ . User latent representation  $\mathbf{z}_u$  is firstly inputted to *interpreter* to infer words that capture user preferences, which is represented by  $g(\mathbf{z}_u)$ , and produce interpreter-based vector  $\hat{\mathbf{z}}_u$ . Final user representation  $\tilde{\mathbf{z}}_u = \mathbf{z}_u + \hat{\mathbf{z}}_u$  is passed to decoder for item recommendation.

Fig. 1. Illustration of INTEREC. Figure (a) presents our general idea while Figure (b) is a realization with *recommender* and *interpreter*.

Table 2. List of Notations

$u, i, j$	User index, item index and word index
$M, N, K$	The number of users, items and words, respectively
$d$	Dimensionality of user, item and word embedding vector
$\epsilon$	Exponential weight of normalization term in memory-based representation (Equation (8))
$\tau$	Temperature hyper-parameter (Equation (6))
$\mathbf{X}, \mathbf{x}_i$	Tf-idf item textual content matrix and textual content vector of item $i$
$\mathbf{R}, \mathbf{r}_u$	User-item interaction matrix and binary rating vector of user $u$
$\mathbf{z}_u, \hat{\mathbf{z}}_u, \tilde{\mathbf{z}}_u$	Latent representation, memory-based representation and combined representation of user $u$
$\mathbf{V}^{text}, \mathbf{V}$	Text-based and collaborative filtering-based item embedding matrices
$\mathbf{M}, \mathbf{K}$	Memory matrix and Key matrix in Memory Network
$\mathbf{a}^T, \mathbf{A}^T$	Transpose of vector (bold lower case letter) and matrix (bold upper case letter)

items, respectively. Each  $u$ th row,  $\mathbf{r}_u$ , of matrix  $\mathbf{R}$  denotes the interaction vector of user indexed by  $u$  and  $\mathbf{r}_{ui} = 1$  indicates interaction between user and item (indexed by  $i$ ). Furthermore, items have side information, i.e., textual content, denoted by matrix  $\mathbf{X} \in \mathbb{R}^{N \times K}$ , with  $K$  is the number of words in vocabulary. Each  $i$ th row  $\mathbf{x}_i$  of  $\mathbf{X}$  is *tf-idf* representation of textual content of item  $i$ . For textual-aware recommendation task, Supervised Learning with Interpretation *SLI* involves



two explicit empirical losses for recommendation, and interpretability.

$$\arg \min_{f \in \mathcal{F}, g \in \mathcal{G}} \mathcal{L}^{rec}(f, \mathbf{R}, \mathbf{X}) + \mathcal{L}^{int}(f, g, \mathbf{R}, \mathbf{X}), \quad (1)$$

in which  $\mathcal{F}$  is the model space of *recommender*,  $\mathcal{G}$  is the model space of *interpreter*,  $\mathcal{L}^{rec}(\cdot)$  is recommendation loss while  $\mathcal{L}^{int}(\cdot)$  is designed for interpretability objective. Next, we describe our proposed realization of SLI called INTEREC, including details of *recommender*  $f$ , *interpreter*  $g$  and learning objectives.

### 3.1 INTEREC

**3.1.1 Recommender.** We set  $\mathcal{F}$  as the class of deep neural networks to learn a recommendation model. Each realization  $f \in \mathcal{F}$  is parameterized by  $\Theta_f$ .  $f$  should satisfy the following properties (i)  $f$  takes user information, e.g., rating vector  $\mathbf{r}_u$  or user's ID, as input and outputs a list of recommended items for user  $u$  and (ii) the output of hidden layer of  $f$  abstractly encodes user's preferences. In this work, we focus on examining two autoencoder-structured variants for  $f$ : vanilla **autoencoder (AE)** and **denoising autoencoder (DAE)**. While other neural recommenders are feasible, the choice of autoencoder reduces model complexity, which will be elaborated later in Section 3.3. We denote the corresponding recommenders are  $f^{AE}$  and  $f^{DAE}$ , respectively.

Encoder  $f_{enc}^{AE}$  of recommender  $f^{AE}$ , which is based on AutoRec [53]:

$$\mathbf{z}_u = f_{enc}^{AE} = e(\mathbf{r}_u \mathbf{W}^{enc} + \mathbf{b}^{enc}). \quad (2)$$

Encoder  $f_{enc}^{DAE}$  of recommender  $f^{DAE}$ , which is based on CDAE [69]:

$$\mathbf{z}_u = f_{enc}^{DAE} = e(\mathbf{r}_u^c \mathbf{W}^{enc} + \mathbf{Q}_u + \mathbf{b}^{enc}). \quad (3)$$

Decoder of recommender  $f^{AE}$  and recommender  $f^{DAE}$  has similar formulation, which is denoted as  $f_{dec}$ :

$$\mathbf{o}_u = f_{dec} = s(\tilde{\mathbf{z}}_u \mathbf{W}^{dec} + \mathbf{b}^{dec}), \quad (4)$$

in which,  $\mathbf{r}_u^c$  is a corrupted version of  $\mathbf{r}_u$ , obtained by randomly zeroing out some elements,  $e$  and  $s$  are non-linearity functions.  $e$  is set to *tanh* for both variants, while  $s$  is set to *sigmoid* for  $f^{AE}$  and *softmax* for  $f^{DAE}$ . These activation functions result in different loss functions for two examined recommendation models, allowing us to test the proposed architecture on different learning scenarios. The difference between  $f_{enc}^{AE}$  and  $f_{enc}^{DAE}$  is that  $f_{enc}^{DAE}$  accepts a corrupted version of user's rating vector and uses bias vector  $\mathbf{Q}_u$ , which is model's parameter, for user representation.  $\tilde{\mathbf{z}}_u = \mathbf{z}_u + \hat{\mathbf{z}}_u$ , where  $\hat{\mathbf{z}}_u$  is *interpreter*-based representation of user  $u$ . Section 3.1.2 describes how to derive  $\hat{\mathbf{z}}_u$ . Parameters of *recommender* is  $\Theta_f = \{\mathbf{W}^{enc} \in \mathbb{R}^{N \times d}, \mathbf{b}^{enc} \in \mathbb{R}^d, \mathbf{W}^{dec} \in \mathbb{R}^{d \times N}, \mathbf{b}^{dec} \in \mathbb{R}^N, \mathbf{Q}_u \in \mathbb{Q} \in \mathbb{R}^{M \times d}\}$ ,  $d$  is the dimensionality. We denote INTEREC with vanilla autoencoder *recommender* is INTEREC-AE while INTEREC-DAE is INTEREC with denoising autoencoder *recommender*.

Both  $f^{AE}$  and  $f^{DAE}$  satisfies the two mentioned properties. Firstly, the output of decoder can be seen as predicted probability of items that user is likely to interact. Secondly,  $\mathbf{z}_u$  can be seen as a compact representation of user's preferences. Its individual dimension, however, is abstract and not immediately interpretable. Therefore, *interpreter* is required to associate these latent features with human-understandable natural language words. Two salient notions in implementing *recommender*  $f$ ,  $f^{AE}$  or  $f^{DAE}$ , are

- To incorporate textual content into *recommender*,  $\mathbf{W}^{dec}$  is implemented as  $\mathbf{W}^{dec} = (\mathbf{V}^{text} + \mathbf{V})^T$ . By doing so, each item is represented by two components,  $\mathbf{V} \in \mathbb{R}^{N \times d}$  is a free matrix learned during training to capture item's features from collaborative filtering signals and  $\mathbf{V}^{text} \in \mathbb{R}^{N \times d}$  captures textual-based item's features and is obtained by stacking outputs of hidden layer in Equation (7). Note that  $\mathbf{V}^{text}$  is left unchanged to preserve its meaning

not to be overwritten by collaborative filtering signals, which is empirically found useful for interpretability as evidenced in Section 4.3. Unlike existing more restrictive models that treat text-based item representation as regularization or to be trained using collaborative filtering signals, this design enables user representation captures both collaborative filtering signals and textual signals more effectively as shown in the experimental results in Section 4.

- For encoder, *tanh* non-linearity is used to model likes and dislikes with positive and negative values, respectively.

**Extension.** To verify the applicability of our proposed method, we examine a recently developed non-autoencoder recommendation model called DirectAU [61]. Under encoder-decoder framework, the encoder of DirectAU is simply a look-up table  $\mathbf{U} \in \mathbb{R}^{M \times d}$  of  $M$  rows, each is vector representation of one user. User  $u$  representation is produced as  $\mathbf{z}_u = \mathbf{U}_u \in \mathbb{R}^d$ . Similarly, item representations are also stored in a look-up table  $\mathbf{V} \in \mathbb{R}^{N \times d}$  of  $N$  rows, each for one item. Item  $i$ 's representation is produced as  $\mathbf{z}_i = \mathbf{V}_i \in \mathbb{R}^d$ . DirectAU distinguishes itself by the learning objective, which will be elaborated in Section 3.3. We name our model variant extending DirectAU as INTEREC-DIRECTAU. INTEREC-DIRECTAU predicts interaction score between user  $u$  and item  $i$  as

$$\mathbf{o}_{ui} = (\tilde{\mathbf{z}}_u)^T \mathbf{W}_i^{dec} = (\mathbf{z}_u + \hat{\mathbf{z}}_u)^T (\mathbf{V}_i^{text} + \mathbf{V}_i). \quad (5)$$

Similar to INTEREC-AE and INTEREC-DAE, user representation in INTEREC-DIRECTAU also contains two terms, one is  $\mathbf{z}_u$  and the other  $\hat{\mathbf{z}}_u$ , which is output of *interpreter* as elaborated in the next section. The interpretation of combined item representation  $\mathbf{W}_i^{dec}$  and  $(\mathbf{V}_i^{text} + \mathbf{V}_i)$  are identical to those of INTEREC-AE and INTEREC-DAE. For INTEREC-DIRECTAU, we empirically found that normalizing each row of  $\mathbf{V}^{text}$  to unit length helps model converge faster and achieve higher accuracy. Additionally,  $\mathbf{V}^{text}$  in INTEREC-DIRECTAU is also left unchanged during training model.

**3.1.2 Interpreter.** Unlike existing interpretability models [1, 8, 51, 73] aiming at interpreting *model prediction* given input as an image or a sentence, our target is interpreting *user's preferences*. In recommender systems, the input is a list of user's adopted items, oftentimes described by their IDs, followed by an embedding layer. Therefore, it is difficult to understand user's preferences based solely on item IDs. As such, the task of *interpreter* is to generate a set of attributes capturing user's preferences. Following [13], our interpretability is formulated as

- Understanding *user's preferences* behind their adoptions.
- The interpretability is evaluated towards how good they capture user's preferences using both *human-grounded metrics* and *functionally-grounded evaluation*.
- The scope of interpretability is *local interpretability*, i.e., understanding preferences of a single user.
- *Single words* from item textual content are treated as *cognitive chunks* or *attributes*, i.e., units of interpretability.

Given user representation  $\mathbf{z}_u$ , which is abstract and not interpretable, *interpreter*  $g, g: \mathbf{z}_u \rightarrow \mathbb{R}^+$ , computes the activation score of user representation with an attribute  $j$ , i.e.,

$$g(\mathbf{z}_u)_j = \text{sigmoid}(\mathbf{z}_u \mathbf{K}_j^T, \tau) = 1 / \left( 1 + e^{-\frac{\mathbf{z}_u \mathbf{K}_j^T}{\tau}} \right), \forall j = 1, 2, \dots, K. \quad (6)$$

$\mathbf{K} \in \mathbb{R}^{K \times d}$  is *key matrix* and also called *dictionary of attributes* in this article. Each row of  $\mathbf{K}$  stores the  $d$ -dimension representation of a word, i.e., cognitive chunk or attribute. Several notions are implemented here.

- *Interpreter*  $g$  accepts user representation from *recommender* as input, enabling interpretation of user's interests.



- Dictionary  $\mathbf{K}$  stores all of  $K$  words in the vocabulary. Consequently,  $g(\mathbf{z}_u)$  is defined over word space, potentially attending to words outside user’s own corpus, resulting a more generalized user’s representation in  $\tilde{\mathbf{z}}_u$  in Equation (4).
- *Sigmoid* non-linearity is used instead of *softmax* as in [46, 56]. The reason is that *softmax* acts as a  $L_1$ -normalization over attributes, overly punishing attentive scores in Equation (6) for active users who are associated with many attributes because of her interactions with a wide range of items. *Sigmoid* allows independent attention over attributes, meaning that many words can have attention score close to 1. In addition, a temperature hyper-parameter,  $\tau$ , is introduced to strengthen the gap between positive and negative elements in  $\mathbf{z}_u \mathbf{K}^T$ . Our finding is consistent with other works [19, 71], which also study method to relax softmax, i.e., output does not sum to 1, to improve recommendation performance.

**Dictionary of Attributes.** A natural question until here is how to derive matrix  $\mathbf{K}$ . Therefore, we seek a function  $\phi : \mathcal{A} \rightarrow \mathbb{R}^d$  that maps each attribute from attribute space  $\mathcal{A}$  to a  $d$ -dimension vector. As in [48],  $\phi$  should encode patterns related to input, which is the list of items in our case. Intuitively, a solution that jointly derives representations of words and items is satisfied. Therefore, we implement  $\phi$  as a **denoising auto-encoder (DAE)** [59], i.e.,

$$\hat{\mathbf{x}}_i = \tanh(\mathbf{x}_i^c \mathbf{K} + \mathbf{b}^{denc}) \mathbf{K}^T + \mathbf{b}^{ddenc}, \quad (7)$$

$\mathbf{x}_i^c$  is the corrupted version<sup>1</sup>  $\mathbf{x}_i$ , which is tf-idf textual content of item  $i$ . Parameters are  $\Theta_1 = \{\mathbf{K} \in \mathbb{R}^{K \times d}, \mathbf{b}^{denc} \in \mathbb{R}^d, \mathbf{b}^{ddenc} \in \mathbb{R}^K\}$ , which are randomly initialized and refined during training. Hence, elements in  $\mathbf{K}$  capture relationships between words, i.e., *cognitive chunks/attributes*, and items. The importance of  $\mathbf{K}$  to the quality of user preference interpretation is analyzed in Section 4.3. Other choices such as CNN [28] or attention [42] are eligible. Finally,  $\tanh(\mathbf{x}_i \mathbf{K} + \mathbf{b}^{denc})$  composes each row of  $\mathbf{V}^{text}$  used in Equation (4).

**Interpretation in INTEREC.** We are interested in interpreting preferences of a single user. This scope is *local interpretability*. The following definition guides our model to output interpretation of user’s preferences.

*Definition 3.1 (Local Interpretability).* A **local interpretation** of user’s preferences for a user  $u$  by an interpreter  $g$  given her representation  $\mathbf{z}_u$  from recommender  $f$  is the set of  $k$  attributes with highest activated scores in Equation (6).

Note that when  $k$ , a pre-chosen number, gets larger, the interpretation is better at covering user’s preferences. From human perspectives, however, large  $k$  of words results in difficulty to quickly grasp user’s preferences.

**Interpretability-Based Representation.** Since  $g(\mathbf{z}_u)$  is defined over word space, it is potential that  $g(\cdot)$  gives higher score for words outside user’s own associated texts. Intuitively, we can generalize user’s representation beyond their interacted items, enabling delivering more interested items to user. We seek a function  $l : \mathbb{R}^K \rightarrow \mathbb{R}^d$  to cater the generalized representation.

$$\hat{\mathbf{z}}_u = l(g(\mathbf{z}_u), \mathbf{M}) = \tanh \left( K^{-\epsilon} \sum_{j=1}^K g(\mathbf{z}_u)_j \mathbf{M}_j \right). \quad (8)$$

Here, each row of value matrix  $\mathbf{M}$ ,  $\mathbf{M}_j \in \mathbb{R}^d$ , stores representation of word  $j$ , which captures collaborative filtering supervision signals.  $\mathbf{M}$  will be trained during learning model.  $K^{-\epsilon}$  is used to promote *conciseness*, one property of interpretability mentioned in [48]. This property expects a small number of attributes for interpretation. We give a detailed explanation for  $\epsilon$  in Section 3.2

<sup>1</sup>Randomly zeroing an element with probability of 0.3.

and empirical study on  $\epsilon$  is presented in Section 4.3. The parameters of *interpreter*  $g$  is  $\Theta_g = \{\mathbf{K}, \mathbf{M}\}$ . After obtaining  $\hat{\mathbf{z}}_u$ , we plug it into Equation (4) or Equation (5).

### 3.2 Model Analysis

Expanding Equation (4) and Equation (5), omitting non-linearity and bias for simplicity, the predicted score between user  $u$  and item  $i$  is

$$\mathbf{o}_{ui} = \tilde{\mathbf{z}}_u \mathbf{W}_i^{dec} = \mathbf{z}_u \mathbf{W}_i^{dec} + \hat{\mathbf{z}}_u \mathbf{W}_i^{dec}. \quad (9)$$

Without the second term  $\hat{\mathbf{z}}_u \mathbf{W}_i^{dec}$  in Equation (9), INTEREC reduces to a form of content-aware recommendation, decoding the adoption-based user encoding  $\mathbf{z}_u$  with item encoding  $\mathbf{W}_i^{dec}$  informed by both content and collaborative filtering signals. Furthermore, by expanding the first term  $\mathbf{z}_u \mathbf{W}_i^{dec}$ , we obtain:

$$\mathbf{z}_u \mathbf{W}_i^{dec} = \mathbf{z}_u (\mathbf{V}_i^{text})^T + \mathbf{z}_u (\mathbf{V}_i)^T. \quad (10)$$

In Equation (10), the first term forces user latent vector  $\mathbf{z}_u$  to capture user preferences from textual content signals, while the second term forces  $\mathbf{z}_u$  to capture user preferences from collaborative filtering signals. By leaving  $\mathbf{V}_i^{text}$  unchanged during training, the textual semantics are preserved. If  $\mathbf{V}_i^{text}$  were to be updated during training, the collaborative filtering signals would potentially change the textual semantics underlying  $\mathbf{V}_i^{text}$ . By separating  $\mathbf{V}^{text}$  and  $\mathbf{V}$ , our model fully exploits the representation power of both content-based representation and collaborative filtering-based representation. In experiments, we empirically show that this plays important role to achieve both of our goals in this article.

In Equation (9), we interpret the second term  $\hat{\mathbf{z}}_u \mathbf{W}_i^{dec}$  as a retrieval function, in which the inferred words act as a query to retrieve relevant items for each user. Expanding  $\hat{\mathbf{z}}_u \mathbf{W}_i^{dec}$  and omitting non-linear activation for simplicity, we have

$$\hat{\mathbf{z}}_u \mathbf{W}_i^{dec} = K^{-\epsilon} \sum_{j=1}^K (g(\mathbf{z}_u)_j \mathbf{M}_j) \mathbf{W}_i^{dec}, \quad (11)$$

$g(\mathbf{z}_u)_j \mathbf{M}_j$  can be interpreted as the  $j$ th word representation w.r.t. user preference on this word.  $g(\mathbf{z}_u)_j \mathbf{M}_j \mathbf{W}_i^{dec}$  measures the similarity between item  $i$  and word  $j$  w.r.t. user  $u$ . The output of Equation (11) is the similarity between item  $i$  and user  $u$  based on the inferred words for  $u$ . If  $g(\mathbf{z}_u)_j$  outputs high score for words outside user's adopted item texts, Equation (8) potentially results in retrieving more relevant items for user.

To understand the role of normalization term  $K^{-\epsilon}$ , we examine two extreme cases. When  $\epsilon = 1$ , the predicted score is averaged over all words in the vocabulary, an item  $i$  gets a high score provided it gets high inner product with nearly all words in the vocabulary. This is unrealistic since each item possesses only a certain number of features, described by their textual content. When  $\epsilon = 0$ , the predicted score is the sum over inner product of all features with item  $i$ . Item  $i$  could get high score if it only gets high inner product with a few words. This may result in retrieving more irrelevant items than relevant items since too few words are insufficient to retrieve relevant items. We believe that an appropriate value of  $\epsilon$  would be somewhere between 0 and 1, which is shown by empirical evidence in Section 4.3.

### 3.3 Learning Objectives

This section elaborates our model's learning objectives for both recommendation and interpretability. We discuss several properties needed to output relevant interpretation as presented in [48].

**Objective Function.** For INTEREC-AE, we use weighted binary cross-entropy loss for optimization

$$\mathcal{L} = -\frac{1}{\mathcal{B}} \sum_{u=1}^{\mathcal{B}} \sum_{i=1}^N \mathbf{C}_{ui} [\mathbf{r}_{ui} \log(\mathbf{o}_{ui}) + (1 - \mathbf{r}_{ui}) \log(1 - \mathbf{o}_{ui})]. \quad (12)$$

For INTEREC-DAE, we empirically find that using cross entropy loss achieves higher accuracy on four chosen datasets.

$$\mathcal{L} = -\frac{1}{\mathcal{B}} \sum_{u=1}^{\mathcal{B}} \sum_{i=1}^N \mathbf{r}_{ui} \log \mathbf{o}_{ui}. \quad (13)$$

$\mathbf{C}_{ui} = 1$  if  $\mathbf{r}_{ui} = 1$ , otherwise  $\mathbf{C}_{ui} = 0.01$  for all datasets, following [32, 62].  $\mathcal{B}$  is batch data size.

For INTEREC-DIRECTAU, the objective includes two terms *alignment* and *uniformity*. While *alignment* encourages the representation of user and the representation of her adopted item are close, *uniformity* encourages discrimination between user and item representations.  $\gamma$  is a hyper-parameter controlling the influence of *uniformity*.

$$\mathcal{L} = \sum_{(u,i) \in \mathcal{B}} \underbrace{E_{u,i \sim p_{pos}} \|f(u) - f(i)\|^2}_{\mathcal{L}_{alignment}} + \gamma \cdot \underbrace{(\log E_{u,u' \sim p_{user}} e^{-2\|f(u)-f(u')\|^2} / 2 + \log E_{i,i' \sim p_{item}} e^{-2\|f(i)-f(i')\|^2} / 2)}_{\mathcal{L}_{uniformity}}, \quad (14)$$

in which  $f(u)$  and  $f(i)$  are unit-length normalization of user representation  $\tilde{\mathbf{z}}_u$  and item representation  $\mathbf{W}_i^{dec}$ , respectively.  $p_{pos}, p_{user}, p_{item}$  are distribution of positive (observed) user-item interactions, users and items, respectively.  $u'$  and  $i'$  are other user and item in the same batch with  $u$  and  $i$ .

Examining multiple variants of *recommender* and their associated learning objectives gives us a broader view of the behavior of our proposed architecture. Minimizing these losses is equivalent to force predicted rating  $\mathbf{o}_u$  for user  $u$  to be closed to ground truth values  $\mathbf{r}_u$ . Compared to the *SLI* framework in Equation (1), our objective includes only one term for both recommendation and interpretability. We now give the explanation for this design as well as examine several other properties discussed in [48] which are encoded Equation (12), Equation (13), and Equation (14).

**Fidelity to Output.** This property requires the *interpreter*  $g$  to be close to *recommender*  $f$ . In [48], the authors impose a regularization for this property by minimizing cross-entropy between outputs of  $g$  and  $f$ , leading to another term in loss function. Here, we implicitly impose this property in Equation (12), Equation (13), and Equation (14). Recall from Equation (4), predicted rating  $\mathbf{o}_u$  is composed of user representation from *recommender*  $f$ , i.e.,  $f^{AE}$  or  $f^{DAE}$ , and *interpreter*  $g$ . Therefore, minimizing Equation (12), Equation (13) and Equation (14) forces  $f$  and  $g$  to converge to the same objective as observed rating  $\mathbf{r}_u$ .

**Conciseness of Interpretation.** A small number of attributes is expected for interpretation because it is easier for human to grasp the user's preferences from generated words. In addition, focusing on smaller number of words implicitly forces model learn to choose illustrative words rather than less representative ones. Our model encodes this property in Equation (8).

**Diversity of Interpretation.** *Diversity* encourages different attributes to be generated given many randomly selected input samples. After learning DAE described in Equation (7),  $\mathbf{K}$  is fixed. Therefore, in Equation (6),  $g(\mathbf{z}_u) \neq g(\mathbf{z}_{u'})$  for  $\mathbf{r}_u \neq \mathbf{r}_{u'}$ . In other words, users with different set of interacted items have different local interpretations. We empirically found that this idea works well and leave other optimization-based method as in [48] for future work. Note that optimization-based methods will introduce new term in loss function, imposing difficulty for convergence. Our design, on the other hand, focuses on only one objective function.

**Fidelity to Input** requires attributes stored in  $\mathbf{K}$  related to input, i.e., the list of adopted items. As elaborated in Section 3.1.2, training denoising autoencoder in Equation (7) inherently captures relationships between words, i.e., attributes, and items. For *recommender*  $f$ , the input is a list of adopted items, hence inherently relates to words stored in  $\mathbf{K}$ . Furthermore, we leverage autoencoder structure for *recommender*  $f$ , enforcing *Fidelity to Output* is equivalent to imposing *Fidelity to Input* for *recommender*, because the input and output of *recommender*  $f$  are both the list of adopted items per user.

Note that FLINT [48] introduces a new loss term for each of the above properties with corresponding hyper-parameters, which is highly dynamical in nature. Finally, let parameters in INTEREC as  $\Theta = \{\Theta_1, \Theta_2\}$ . In which,  $\Theta_1 = \{\mathbf{K}, \mathbf{b}^{dec}, \mathbf{b}^{dec}\}$  are parameters of denoising autoencoder in Equation (7) while  $\Theta_2 = \{\Theta_f, \Theta_g\}$  are parameters of *recommender*  $f$  and *interpreter*  $g$ . Learning model boils down to learning its parameters  $\Theta$ . Our training procedure including two stages. In the first stage, we train denoising autoencoder in Equation (7) with  $\Theta_1$  and then fix these parameters regarding use or not use in the second stage. Then, we train  $\Theta_2$  in the second stage with loss term defined in Equation (12). Algorithm 1 presents more details the training procedure of our proposed model.

## 4 EXPERIMENTS

Our experiments seek to evaluate both aspects of recommendation accuracy and interpretability of user preferences. We aim at answering the following research questions

- (RQ1) How does the proposed model INTEREC, which consists of *recommender* and *interpreter*, perform recommendation compared to existing baselines, including collaborative filtering and textual-aware models?
- (RQ2) Is the *interpreter* in INTEREC able to generate textual units which well capture user’s preferences? How does *interpreter* affect the recommendation performance?
- (RQ3) What are the effects of key components in *recommender* and *interpreter* on recommendation performance as well as the interpretation of user’s preferences?

**Datasets.** We consider *CDs & Vinyl, Cell Phones, Toys & Games*, which are three categories of the *Amazon* dataset [17, 44]. We use the provided 5-core data<sup>2</sup> where each user and item has a least five reviews. For each item, we concatenate its title and description, and the resulting text is referred to as item’s textual description/content. *Citeulike-a*<sup>3</sup>[62]. To branch out to a non-product dataset, we use dataset that associates users and academic articles. *Item’s textual description* is the concatenation of title and abstract of each article.

**Preprocessing.** We first remove html code (if any) then employ spaCy [21] to tokenize text into single words. For each dataset, we only keep words with frequency higher than 5 and appearing in less than 50% of textual descriptions, remove all stop words and retain most frequent words as the vocabulary proportionately to the dataset size. All items for which textual description is empty, i.e., there is no in-vocabulary word in the description, together with their interactions with users are discarded. Table 3 shows the statistics of our data after preprocessing.

**Data Split.** We construct training, validation and testing set by utilizing leave-one-out strategy following [20]. For *Amazon* datasets with provided time-stamp, we first sort the user’s interactions chronologically. Then the latest item of each user is added to test set, the penultimate is added to validation set and the remaining items are added to the training set. For *citeulike-a* dataset, where timestamp of each user-article interaction is not available, for each user, we select a random article for the validation set and the test set respectively.

<sup>2</sup><https://jmcauley.ucsd.edu/data/amazon/>

<sup>3</sup><http://wanghao.in/CDL.htm>

Table 3. Statistics of Dataset used in our Experiments

Data	#users	#items	#interactions	#words
Cell Phones	4,775	4,883	31,749	4,000
Toys & Games	9,466	7,964	80,919	8,000
CDs & Vinyl	38,406	39,260	524,459	10,000
Citeulike-a	5,551	16,980	204,986	8,000

**ALGORITHM 1:** Training procedure of INTEREC**Input:**

- *tf-idf*-based text matrix of items  $\mathbf{X} \in \mathbb{R}^{N \times K}$
- rating matrix  $\mathbf{R} \in \{0, 1\}^{M \times N}$ ,  $\mathbf{r}_u \in \{0, 1\}^N$  is binary rating vector of user  $u$
- model’s collective parameters  $\Theta = \{\Theta_1, \Theta_2\}$ 
  - parameters of denoising autoencoder for modeling item textual content  $\Theta_1 = \{\mathbf{K}, \mathbf{b}^{denc}, \mathbf{b}^{ddec}\}$
  - parameters of INTEREC  $\Theta_2 = \{\Theta_f, \Theta_g\}$ 
    - \* parameters of *recommender*  $\Theta_f = \{\mathbf{W}^{enc}, \mathbf{b}^{enc}, \mathbf{W}^{dec}, \mathbf{b}^{dec}, \mathbf{V}\}$  or  $\{\mathbf{U}, \mathbf{V}\}$  (for INTEREC-DIRECTAU)
    - \* parameters of *interpreter*  $\Theta_g = \{\mathbf{M}, \mathbf{K}\}$

**Output:** updated  $\Theta$ 

```

1 Randomly initialize model’s parameters
  // Stage 1: training denoising autoencoder for item textual content modeling
2 for batch item  $\mathcal{B}_{item}$  do
3   for  $i \in \mathcal{B}_{item}$  do
4      $\hat{\mathbf{x}}_i = \tanh(\mathbf{x}_i^c \mathbf{K} + \mathbf{b}^{denc}) \mathbf{K}^T + \mathbf{b}^{ddec}$ 
5      $\mathcal{L}_{item} = \frac{1}{|\mathcal{B}_{item}|} \sum_{i \in \mathcal{B}_{item}} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2$ 
6     Update  $\Theta_1$  to minimize  $\mathcal{L}_{item}$ 
7 Calculate  $\mathbf{V}_i^{text} = \tanh(\mathbf{x}_i \mathbf{K} + \mathbf{b}^{denc})$ ,  $\forall i = 1, 2, \dots, N$ ,  $\mathbf{V}^{text} \in \mathbb{R}^{N \times d}$ 
8 Use  $\mathbf{V}^{text}$  and  $\mathbf{K}$  in stage 2
  // Stage 2: training recommender and interpreter in INTEREC.
9 for batch user  $\mathcal{B}_{user}$  do
10  for  $u \in \mathcal{B}_{user}$  do
11     $\mathbf{z}_u \leftarrow$  Output of encoder as Equation (2), Equation (3) or  $\mathbf{U}_u$  (for INTEREC-DIRECTAU)
12     $\hat{\mathbf{z}}_u \leftarrow$  Output of interpreter as in Equation (8)
13     $\tilde{\mathbf{z}}_u = \mathbf{z}_u + \hat{\mathbf{z}}_u$  // Final user representation.
14     $\mathbf{o}^u \leftarrow$  Interaction prediction as Equation (4) or Equation (5)
15 Calculate objective  $\mathcal{L}_{user} \leftarrow$  Equation (12), Equation (13) or Equation (14)
16 Update  $\Theta_2$  to minimize  $\mathcal{L}_{user}$  //  $\mathbf{V}^{text}$  and  $\mathbf{K}$  are not updated, as shown by empirical study in
    Section 4.3

```

**Baselines.** We compare INTEREC against a series of baselines, including collaborative filtering and text-aware recommendation baselines on implicit feedback data.

**Collaborative filtering models:**

- **AutoRec** [53] uses autoencoder for collaborative filtering. INTEREC-AE’s *recommender* is based on AutoRec.
- **CDAE** [69] studies recommendation problem through autoencoder view and learns to recommend items from corrupted inputs. The *recommender* in INTEREC-DAE is based on CDAE.

- **NeuMF** [20] combines generalized matrix factorization, which linearly models user/item latent feature interactions, and multi-layer perceptron to learn the interaction function between users and items.
- **LightGCN** [18] improves Graph Convolutional Network for collaborative filtering by using linear propagation and weighted sum of multi-layered embeddings.
- **ENMF** [7] proposes to learn a neural matrix factorization-based recommendation model without sampling via reformulating the loss function.
- **DirectAU** [61] improves collaborative filtering by optimizing uniformity and alignment of user and item representations. The *recommender* in INTEREC-DIRECTAU is based on DirectAU.

#### Text-aware recommendation models:

- **CDL** [62] proposes a probabilistic model that jointly learns **Stacked Denoising Autoencoder (SDAE)** for text modeling and collaborative filtering.
- **CVAE** [32] presents a similar approach with CDL but replacing SDAE by **Variational Autoencoder (VAE)**.
- **GATE** [42] leverages attention to model textual content and neighbor information to enrich item representation.
- **JSR** [74] jointly predicts user-item interactions and reconstructs textual content.

As we aim at interpreting *user's preferences*, we involve *user-oriented* baselines, i.e., incorporating textual content from user's corpus on user side. For recommendation, we compare INTEREC-AE, INTEREC-DAE and INTEREC-DIRECTAU with both user-oriented and item-oriented competitors. Regarding interpretability, only user-oriented baselines are comparable with ours because they are able to generate a set of words representing user's interests. For fair comparison, all models use the same word vocabulary with our proposed model.

**Model Training.** We use Nvidia Quadro RTX 8000 GPU machines for training with Adam optimizer [29]. Learning rate is chosen from {0.0003, 0.001, 0.003, 0.005} and dropout rate is chosen from {0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.6}. In INTEREC-AE, a dropout layer is added over  $\hat{z}^u$  before decoding step. For other baselines, we follow the original architecture to place dropout layer. For proposed models, the maximum number of training epochs is set to 200. Training stops after 12 epochs without improving HR@20 on validation set. The average results over 10 runs with different random seeds are reported.

**Hyper-parameters** are first chosen based on validation set, then we re-train models with chosen ones and report results on the test set.  $d = 50$  is set for Citeulike-a and Cell Phones datasets and  $d = 100$  for others.  $a = 1$  and  $b = 0.01$  are weights for observed and unobserved interactions, respectively, in CDL, CVAE, and INTEREC-AE. For fair comparison, we initialize word embedding matrix in JSR<sup>4</sup> as  $\mathbf{K}$ . Table 4 presents more details of the search space of hyper-parameters. Given the search space of hyper-parameters, we employ grid search approach for baselines and INTEREC to choose the set of hyper-parameters achieving the best recommendation accuracy, i.e., HR@20, on validation set. Then we re-train all models with chosen hyper-parameters and report their performance on test set.

**Metrics.** Hit Ratio at top- $k$  (HR@K) and **Normalized Discounted Cumulative Gain at top- $k$**  (NDCG@K) are employed for recommendation and retrieval-based interpretability evaluation.

$$HR@K = \frac{1}{|\mathcal{U}|} \sum_u \mathbb{1}[\delta(R(u) \cap T(u) \neq \emptyset)].$$

<sup>4</sup>In original article, authors leverage relevance-based word embeddings, which is not available in our work.



Table 4. Search Space for Hyper-parameters in Baselines and the Proposed INTEREC

Model	Search Space
AutoRec	activation function $\in \{sigmoid, relu, tanh\}$ ; weight decay $\in \{0.0001, 0.001, 0.01\}$
CDAE	activation function $\in \{sigmoid, relu, tanh\}$ ; weight decay $\in \{0.0001, 0.001, 0.01\}$ corruption ratio $\in \{0.1, 0.3, 0.5\}$
NeuMF	hidden layer size $\in \{8, 16, 32, 64, 128, 150, 200\}$ ; number of layers $\in \{2, 3\}$ ; dropout $\in \{0.1, 0.3, 0.5\}$
LightGCN	number of layers $\in \{1, 2, 3\}$ ; $L_2$ regularization coefficient $\in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$
ENMF	dropout $\in \{0.3, 0.5, 0.7, 0.9\}$ ; weight of missing data $c_0 \in \{0.005, 0.01, 0.05, 0.1, 0.2, 0.5, 1\}$
DirectAU	$\gamma \in \{0.1, 0.2, 0.5, 1, 2, 5\}$ ; weight decay $\in \{0, 10^{-4}, 10^{-6}, 10^{-8}\}$ ; learning rate in $\{0.001, 0.003, 0.004\}$
CDL	$\lambda_u \in \{10^{-x}, x \in 6, 5, 4, 3, 2, 1, 0\}$ ; $\lambda_v \in \{10^{-x}, x \in 6, 5, 4, 3, 2, 1, 0\}$ ; $\lambda_r = 1$ ; $\lambda_n = 10^3$ $\lambda_w = 10^{-4}$ for Amazon subsets and $\lambda_w = 10^{-1}$ for Citeulike-a
CVAE	$\lambda_u \in \{10^{-x}, x \in 6, 5, 4, 3, 2, 1, 0\}$ ; $\lambda_v \in \{10^{-x}, x \in 6, 5, 4, 3, 2, 1, 0\}$ ; $\lambda_r = 1$ ; $\lambda_n = 10^3$ $\lambda_w = 10^{-4}$ for Amazon subsets and $\lambda_w = 10^{-1}$ for Citeulike-a
GATE	maximal sequence length = 4000; $\rho \in \{20, 50, 100\}$ ; $d_a \in \{10, 20, 50\}$ ; hidden layer size $\in \{100, 200\}$
JSR	$\eta = 4$ ; hidden layer size $\in \{50, 100, 200\}$ ; weight decay = $10^{-3}$ $\lambda = 0.5$ for Amazon subsets and $\lambda_w = 0.1$ for Citeulike-a
INTEREC	weight decay = $10^{-4}$ ; $\tau \in \{0.01, 0.02\}$ ; $\epsilon$ is studied in Section 4.3 corruption ratio $\in \{0.1, 0.3, 0.5\}$ (for INTEREC-DAE variant) $\gamma \in \{0.1, 0.2, 0.5, 1, 2, 5\}$ ; weight decay $\in \{0, 10^{-4}, 10^{-6}, 10^{-8}\}$ ; learning rate in $\{0.001, 0.003, 0.004\}$ (for INTEREC-DIRECTAU variant)

For INTEREC, more extensive hyper-parameter analysis is presented in Section 4.3.

$R(u), T(u)$  are the set of predicted items and test items of user  $u$ , respectively.  $\mathbb{1}[x]$  returns 1 if  $x$  is true, otherwise is 0.

$$NDCG@K = \frac{1}{|U|} \sum_u \frac{DCG@K}{IDCG@K} \quad DCG@K = \sum_r^K \frac{2^{rel_r} - 1}{\log_2(r + 1)}.$$

IDCG@K is the biggest possible value of DCG@K, obtained by treating sorted test set as a length-K prediction.  $rel_r = 1$  indicates relevance, otherwise is 0.

#### 4.1 Recommendation Evaluation

Table 5 reports top-N recommendation performance. We have the following observations.

- Three variants of INTEREC, INTEREC-AE, INTEREC-DAE and INTEREC-DIRECTAU, enjoy significantly better recommendation performance than all baselines four chosen datasets. This supports our design of joint learning *recommender* and *interpreter*, i.e., the design of decoder (Section 3.1) as well as the incorporation of *interpreter* (Section 3.1.2), do not sacrifice recommendation accuracy. Rather, this approach brings performance gain on chosen datasets.
- For autoencoder-based variants, the performance of INTEREC is based on the power of *recommender*. Contrasting the numbers of INTEREC and those of AutoRec and CDAE, it can be seen that the relative performance comparison between AutoRec and CDAE, reflects similar comparison between INTEREC-AE and INTEREC-DAE. For example, on Cell Phones, AutoRec is better than CDAE and INTEREC-AE is also better than INTEREC-DAE. Regarding INTEREC-DIRECTAU, this also applies on Toys & Games and CDs & Vinyl and some specific metrics on the other two datasets.
- *Textual content is an important factor to improve recommendation performance* on chosen datasets. Firstly, this is evidenced by contrasting the numbers of AutoRec, CDAE, DirectAU with those of INTEREC, which extends AutoRec, CDAE and DirectAU by incorporating textual content informed by item representations (in decoder for autoencoder-based variants) and *interpreter*. Secondly, our variants INTEREC-AE and INTEREC-DAE achieve

Table 5. Recommendation Performance Comparison

Dataset	Metric	Collaborative filtering					Item-oriented Fashion				User-oriented Fashion				INTEREC			
		Auto-Rec	CDAE	NeuMF	LightGCN	ENMF	Direct-AU	CDL	CVAE	JSR	GATE	CDL	CVAE	JSR	GATE	AE	DAE	Direct-AU
Cell Phones	H@20	8.55	8.40	6.99	9.43	8.20	<u>9.65</u>	9.52	8.73	8.79	7.37	9.58	8.43	9.14	8.17	<b>10.57</b> <sup>†</sup>	<b>10.34</b>	<u>9.86</u>
	N@20	3.84	3.78	3.05	4.10	3.69	4.03	4.06	3.75	3.93	3.12	<u>4.12</u>	3.63	4.00	3.56	<b>4.46</b> <sup>†</sup>	<b>4.35</b>	<u>4.10</u>
	H@50	14.34	13.89	11.79	15.58	13.36	<u>16.54</u>	15.80	14.37	14.77	12.67	15.76	14.23	15.05	14.02	<b>17.78</b> <sup>†</sup>	<b>17.05</b>	<u>17.39</u>
	N@50	4.98	4.87	4.00	5.31	4.71	<u>5.39</u>	5.30	4.87	5.11	4.17	5.34	4.78	5.16	4.70	<b>5.88</b> <sup>†</sup>	<b>5.68</b>	<u>5.58</u>
Toys & Games	H@20	6.80	6.86	6.03	7.25	6.72	<u>7.39</u>	7.15	7.01	6.99	5.49	7.16	6.87	6.69	5.36	<b>9.35</b>	<u>9.24</u>	<b>9.49</b> <sup>†</sup>
	N@20	2.96	2.98	2.61	3.07	2.87	<u>3.13</u>	3.01	3.00	2.91	2.37	3.02	2.91	2.79	2.29	<b>3.85</b>	<b>3.90</b>	<b>3.96</b> <sup>†</sup>
	H@50	11.13	11.08	9.87	11.99	10.66	<u>12.30</u>	11.66	11.44	11.25	8.98	11.78	11.35	11.10	9.02	<b>15.39</b>	<b>15.38</b>	<b>15.41</b> <sup>†</sup>
	N@50	3.82	3.81	3.37	4.00	3.65	<u>4.10</u>	3.90	3.88	3.75	3.06	3.93	3.69	3.66	3.01	<b>5.05</b>	<b>5.11</b>	<b>5.13</b> <sup>†</sup>
CDs & Vinyl	H@20	8.80	9.62	7.00	9.02	8.66	<u>10.30</u>	7.54	9.10	5.88	7.70	7.70	8.83	5.86	7.16	<u>9.36</u>	<b>9.56</b>	<b>10.55</b> <sup>†</sup>
	N@20	3.71	4.12	2.92	3.83	3.65	<u>4.39</u>	3.17	3.81	2.40	3.25	3.25	3.75	2.38	3.01	<u>3.97</u>	<b>4.10</b>	<b>4.52</b> <sup>†</sup>
	H@50	14.48	15.59	11.96	14.80	14.20	<u>16.63</u>	12.67	15.06	10.30	12.79	12.69	14.54	10.61	11.95	<u>15.27</u>	<b>15.56</b>	<b>16.83</b> <sup>†</sup>
	N@50	4.84	5.30	3.90	4.97	4.75	<u>5.65</u>	4.18	4.99	3.27	4.25	4.23	4.88	3.32	3.95	<u>5.14</u>	<b>5.29</b>	<b>5.76</b> <sup>†</sup>
Citeulike-a	H@20	19.78	25.94	20.47	23.07	20.33	<u>27.48</u>	20.10	21.11	19.94	23.23	20.24	20.93	19.75	22.55	<u>24.94</u>	<b>28.47</b> <sup>†</sup>	<b>28.31</b>
	N@20	8.80	12.33	9.06	10.49	9.17	<u>13.04</u>	9.07	9.38	8.17	10.50	8.95	9.23	8.06	10.50	<u>11.29</u>	<b>13.83</b> <sup>†</sup>	<b>13.39</b>
	H@50	31.34	37.70	32.46	35.56	31.85	<u>40.63</u>	32.46	33.53	32.74	35.27	32.59	33.18	32.82	33.39	<u>38.10</u>	<b>41.01</b>	<b>41.40</b> <sup>†</sup>
	N@50	11.08	14.66	11.44	12.95	11.45	<u>15.64</u>	11.51	11.84	10.70	12.89	11.38	11.66	10.65	12.65	<u>13.89</u>	<b>16.31</b> <sup>†</sup>	<b>15.98</b>

*Item-oriented fashion* baselines equip textual content on item side while *User-oriented fashion* baselines incorporate texts on user side. Among baseline models, the highest number is double underlined. Regarding our proposed variants, the highest number is **boldfaced**, the first runner-up is **boldfaced** and underlined while the second runner-up is underlined. <sup>†</sup> denotes statistical significance between the boldfaced and the double underlined on paired t-test with p-value < 0.01. H@K and N@K stand for Hit Ratio at top K and Normalized Discounted Cumulative Gain at top K. Number unit is percentage (%).

higher recommendation accuracy than LightGCN and CDL, despite AutoRec and CDAE, which are the base of the recommenders, are worse than LightGCN and CDL. Similarly, INTEREC-AE and INTEREC-DAE are also better than DirectAU in 3 out of 4 datasets given that AutoRec and CDAE are worse than DirectAU. Thirdly, textual-aware recommendation models CDL, CVAE and JSR, item-oriented or user-oriented fashion, generally works better than collaborative filtering counterparts, e.g., AutoRec, NeuMF, ENMF. GATE does not work well on Cell Phones and Toys & Games, which we conjecture that it stems from noisy and very long text sequences that GATE processes.

- On CDs & Vinyl, *the importance of textual content is model-dependent*. While textual content is helpful to improve AutoRec and DirectAU, it shows a slightly negative effect when incorporating into CDAE. This suggests careful design and inspection when applying our proposed approach on different *recommenders*.
- Among baselines, DirectAU stands out, achieving higher recommendation accuracy than all other baselines w.r.t. chosen metrics on four datasets, except NDCG@20 on Cell Phone. This is explained by the design of learning objective of DirectAU, which has been shown to be powerful for collaborative filtering. Next, LightGCN generally performs well across datasets. This is attributed to the high order connection modeling in LightGCN. While there is a small gap between performance of CDAE and AutoRec on Cell Phones and Toys & Games and CDs & Vinyl, this performance gap is much bigger on Citeulike-a. We empirically found that cross entropy loss used when training CDAE helps to achieve favorable performance on some specific datasets.

Table 6. Recommendation Performance when Removing *Interpreter* and  $V^{text}$  from INTEREC

Dataset	Metric	INTEREC-AE			INTEREC-DAE			INTEREC-DIRECTAU		
		full model	without <i>interpreter</i>	without $V^{text}$	full model	without <i>interpreter</i>	without $V^{text}$	full model	without <i>interpreter</i>	without $V^{text}$
Cell Phones	H@20	<b>10.57</b>	10.33	8.96	<b>10.34</b>	10.00	8.73	9.86	9.58	<b>10.10</b>
	N@20	<b>4.46</b>	4.40	3.99	<b>4.35</b>	4.21	3.91	4.10	3.83	<b>4.30</b>
	H@50	<b>17.78</b>	17.54	14.80	<b>17.05</b>	16.77	14.43	<b>17.39</b>	16.90	17.34
	N@50	<b>5.88</b>	5.83	5.14	<b>5.68</b>	5.54	5.03	5.58	5.27	<b>5.72</b>
Toys & Games	H@20	<b>9.35</b>	9.09	6.80	9.24	<b>9.37</b>	6.50	9.49	<b>9.50</b>	7.43
	N@20	<b>3.85</b>	3.78	2.95	3.90	<b>3.94</b>	2.84	3.96	<b>3.97</b>	3.14
	H@50	<b>15.39</b>	15.13	10.98	<b>15.38</b>	15.03	10.65	15.41	<b>15.44</b>	12.00
	N@50	<b>5.05</b>	4.98	3.77	<b>5.11</b>	5.06	3.66	5.13	<b>5.14</b>	4.04
CDs & Vinyl	H@20	<b>9.36</b>	9.09	9.10	<b>9.56</b>	9.45	<b>9.56</b>	10.55	<b>10.56</b>	10.49
	N@20	<b>3.97</b>	3.82	3.87	<b>4.10</b>	4.03	4.07	<b>4.52</b>	<b>4.52</b>	4.50
	H@50	<b>15.27</b>	15.05	14.89	<b>15.56</b>	15.45	15.50	16.83	<b>16.85</b>	16.65
	N@50	<b>5.14</b>	5.00	5.01	<b>5.29</b>	5.21	5.25	<b>5.76</b>	<b>5.76</b>	5.72
Citeulike-a	H@20	<b>24.94</b>	22.60	23.45	<b>28.47</b>	28.28	26.27	<b>28.31</b>	28.13	27.69
	N@20	<b>11.29</b>	10.10	10.69	<b>13.83</b>	13.61	12.53	<b>13.39</b>	13.37	13.22
	H@50	<b>38.10</b>	35.72	35.30	<b>41.01</b>	40.88	37.60	<b>41.40</b>	41.05	40.85
	N@50	<b>13.89</b>	12.69	13.04	<b>16.31</b>	16.10	14.78	<b>15.98</b>	15.94	15.83

Unit of numbers is percentage (%).

In what follows, we further analyze the performance change w.r.t. the presence of  $V^{text}$  and *interpreter* in Table 6.

- For autoencoder-based variants, INTEREC-AE and INTEREC-DAE, on Cell Phones, Toys & Games and Citeulike-a, a significant performance degradation is observed when  $V^{text}$  is not present. This suggests textual content is the key factor to resolve sparsity in order to achieve better recommendation accuracy. On CDs & Vinyl where textual content is not an important factor for user-item interactions, it is observed that textual content has slight positive effect on INTEREC-AE and INTEREC-DAE.
- For INTEREC-DIRECTAU, we observe the same trend on Toys & Games, CDs & Vinyl and Citeulike-a. Interestingly, when removing  $V^{text}$  on Cell Phone, we observe a slight increase in model performance. This might stem from the learning objective of INTEREC-DIRECTAU, where the normalized item textual representation does not align well with normalized collaborative filtering item representation.
- When removing *interpreter* from INTEREC-AE, we observe a small performance drop on Cell Phones, Toys & Games, CDs & Vinyl and a significant degradation on Citeulike-a. This shows that *interpreter* is able to generalize user representation to bring performance gain, particularly on Citeulike-a dataset. For INTEREC-DAE, we also observe the negative effect when removing *interpreter* on the majority of datasets. A special case is on Toys & Games where removing *interpreter* has positive effect on top 20 but negative effect on top 50 metrics. This suggests that *interpreter* output helps to discover more relevant items to user but rank them lower on the list. Regarding INTEREC-DIRECTAU, while *interpreter* has tiny influence on recommendation accuracy on Toys & Games and CDs & Vinyl, it is clearer that *interpreter* boosts the recommendation performance on Cell Phones and Citeulike-a.
- Contrasting the numbers of INTEREC without  $V^{text}$  in Table 6, meaning that the model includes *interpreter*, and those of AutoRec, CDAE and DirectAU in Table 5, we find that *interpreter* actually brings performance gain without the presence of  $V^{text}$  on Cell Phones

and Citeulike-a, CDs and Vinyl. This is supporting evidence for the generalization brought by *interpreter* on specific datasets. When both *interpreter* and  $\mathbf{V}^{text}$  present in our unified INTEREC, the gap between INTEREC and AutoRec, CDAE and DirectAU is further enlarged, showing that textual content is not only helpful for recommendation performance but also beneficial for *interpreter*, i.e., assists *interpreter* to discover relevant words for higher recommendation performance.

- The intuition behind *interpreter* is what follows. Existing works on interpretability, e.g., FLINT [48], often consider the tradeoff between accuracy and interpretability since these models force the output of *recommender* (or *predictor*) close to that of *interpreter*. This design is less effective as *interpreter* might not be good at performing target task of *recommender*. To resolve, we design *interpreter* to predict the same target with *recommender*, which reinforces the ability of *interpreter* in performing target task (recommendation in our case). Hence, *interpreter* is not only good at interpretability but also recommendation. Additionally, our proposal of using key-value memory network generalizes user representation by attending to words outside user’s interacted item content. These beyond user interacted text words are able to retrieve relevant items that users are interested in yet have not interacted, leading to better accuracy.

## 4.2 Interpretability Evaluation

For interpretability evaluation, we focus on INTEREC-AE as this variant works well across datasets and to keep this article focused on interpretability evaluation. We closely follow [13] to design interpretability evaluation. Two types of evaluation are applicable to our case, namely *human-grounded metric* and *functionally-grounded evaluation*.

**4.2.1 Human-grounded Evaluation.** The goal is to conduct a simpler experiment that maintains the essence of target application [13], which is recommender system in this article. Since we leverage words as means of interpretation, the experiment should reflect how human comprehends a user’s list of adopted items and match their comprehension with words. Therefore, we engage 10 participants, who are not the authors of this article and are not aware of the research objectives. Among those, some have **Computer Science (CS)** background while others are non-CS major. In this article, we refer *participants* as humans who help us judge the quality of generated words while *users*, without other specification, are from chosen datasets. We randomly select 20 users from 4 chosen datasets. For each user, we collect the list of their adopted items’ titles, which are short sentences describing main content/feature of items. A set of 30 words, which are the outputs of INTEREC-AE, GATE and JSR<sup>5</sup>, are coupled with each user’s associated list of titles. Each participant would go through the list of titles and the set of generated words of each user. Participants would choose any word(s) that they comprehend to understand the list of titles. We regard participants’ choices as ground truth and generated words from each model as prediction. Let  $\mathbf{D}^g$  and  $\mathbf{D}^p$  is the list of ground truth words and predicted words for each user, respectively. Metrics are **Precision (PR)**, **Recall (RE)**, and **Mean Reciprocal Rank (MR)**.

$$\mathbf{PR} = \frac{|\mathbf{D}^g \cap \mathbf{D}^p|}{|\mathbf{D}^p|}; \quad \mathbf{RE} = \frac{|\mathbf{D}^g \cap \mathbf{D}^p|}{|\mathbf{D}^g|}; \quad \mathbf{MR} = \frac{1}{|\mathbf{D}^p|} \sum_{w \in \mathbf{D}^p} \frac{\mathbb{1}[w \in \mathbf{D}^g]}{\mathit{rank}^{\mathbf{D}^p}(w)}, \quad (15)$$

in which  $|\cdot|$  is the *cardinality*,  $\mathit{rank}^L(w)$  returns the rank of element  $w$  in list  $L$  and  $\mathbb{1}[x]$  returns 1 if  $x$  is true, 0 otherwise. The reported numbers are calculated per participant, averaged over 20 samples,

<sup>5</sup>These models represent major approaches in existing content-aware recommendation models, namely regularization (JSR), attention mechanism (GATE) and memory network-based interpreter (ours).

Table 7. *Human-grounded* Interpretability Evaluation

Metric	Model	Participant										Avg.
		1	2	3	4	5	6	7	8	9	10	
PR	GATE	<u>11.50</u>	11.00	11.50	<u>14.00</u>	<u>19.00</u>	<u>57.50</u>	<u>29.00</u>	<u>5.00</u>	<b>28.50</b>	<u>15.50</u>	<u>20.25</u>
	JSR	<u>10.50</u>	<u>13.50</u>	<u>15.50</u>	13.50	17.50	28.00	14.50	1.50	12.50	14.50	14.15
	INTEREC	<b>14.00</b>	<b>25.00<sup>†</sup></b>	<b>23.50<sup>†</sup></b>	<b>27.50<sup>†</sup></b>	<b>32.50<sup>†</sup></b>	<b>71.00<sup>†</sup></b>	<b>34.00</b>	<b>6.00</b>	<b>28.50</b>	<b>23.00</b>	<b>28.50<sup>†</sup></b>
RE	GATE	<u>49.58</u>	34.08	38.42	27.93	36.03	<u>45.33</u>	<u>47.96</u>	<u>50.00</u>	<b>54.60</b>	<b>56.54</b>	<u>44.05</u>
	JSR	42.08	<u>44.92</u>	45.75	30.76	36.40	23.83	24.42	15.00	24.67	33.19	32.10
	INTEREC	<b>54.58</b>	<b>76.08<sup>†</sup></b>	<b>72.17<sup>†</sup></b>	<b>68.24<sup>†</sup></b>	<b>70.84<sup>†</sup></b>	<b>57.93<sup>†</sup></b>	<b>57.14</b>	<b>60.00</b>	<u>52.28</u>	<u>50.12</u>	<b>61.94<sup>†</sup></b>
MR	GATE	2.28	3.29	3.17	3.99	4.65	<u>16.15</u>	<u>9.30</u>	<b>1.86</b>	<b>9.34</b>	4.04	5.81
	JSR	<u>4.91</u>	5.53	6.31	<u>6.05</u>	6.75	10.12	5.94	0.24	3.78	<u>6.22</u>	5.58
	INTEREC	<b>5.29</b>	<b>9.60<sup>†</sup></b>	<b>8.70<sup>†</sup></b>	<b>9.72<sup>†</sup></b>	<b>10.90<sup>†</sup></b>	<b>21.10<sup>†</sup></b>	<b>10.57</b>	<u>1.68</u>	<u>7.97</u>	<u>7.49</u>	<b>9.30<sup>†</sup></b>

Reported numbers per participant are averaged over 20 samples. Bold numbers are the best results while the runner-up is underlined. <sup>†</sup> denotes statistically significant w.r.t. to second best number on a paired t-test with p-value < 0.05. Unit of reported numbers is percentage (%).

as shown in Table 7. Evidently, our proposed model outperforms GATE and JSR convincingly, which is also the consensus among the majority of participants.

**4.2.2 Functionally-grounded Evaluation.** Since human-grounded evaluation is costly, we also seek functionally-grounded evaluation. This method requires a formal definition of interpretability as proxy for quality evaluation [13]. Since we characterize interpretability of user’s preferences using words from items’ content, these words can be intuitively employed as means to retrieve items that fit user’s needs. Hence, we formalize the proxy as a *retrieval* task, i.e., generated words are used to form a query to retrieve items based on the similarity between query and items’ textual content. Top items with highest similarity score are presented for each user.

The *retrieval* task involves query  $\mathbf{q}$ , document  $\mathbf{D}$  and retrieval function  $h$ . Query  $\mathbf{q}$  consists of 10 *cognitive chunks*, i.e., single words, generated by each model while textual description of each item is treated as document  $\mathbf{D}$ . For user-oriented CDL, CVAE and JSR, 10 words with highest predicted scores from user’s text modeling component are taken to form  $\mathbf{q}$ . These words intuitively reflect user’s preferences since their predicted score is based on textual representation, which is a regularization of user’s representation. For user-oriented GATE,  $\mathbf{q}$  is created from 10 words with highest attentive scores, following original article. For INTEREC, we follow definition 3.1 to create  $\mathbf{q}$  with  $k = 10$ . More values of  $k$  will be studied in Table 9. Retrieval function is  $h(\mathbf{q}, \mathbf{D}) = \text{agg}_{w \in \mathbf{D}} \max_{w' \in \mathbf{q}} \frac{\langle \mathbf{e}^{w'}, \mathbf{e}^w \rangle}{|\mathbf{e}^{w'}| \cdot |\mathbf{e}^w|}$ ,  $w$  and  $w'$  are words in document  $\mathbf{D}$  and query  $\mathbf{q}$ , respectively.  $\mathbf{e}^w$  and  $\mathbf{e}^{w'}$  are embeddings of word  $w$  and  $w'$ .  $\max_{w' \in \mathbf{q}} \frac{\langle \mathbf{e}^{w'}, \mathbf{e}^w \rangle}{|\mathbf{e}^{w'}| \cdot |\mathbf{e}^w|}$  follows Equation (2) in [27], which measures the semantic similarity between term  $w$  with respect to short text  $\mathbf{q}$ . By leveraging distributed vector representation [45], the vocabulary mismatch problem is alleviated. *agg* is an aggregation function.<sup>6</sup> Since our method and competitors have different notions of word embeddings, we leverage Word2Vec in Gensim [49] to obtain word embeddings. Therefore, the output of retrieval function is not biased towards any competitors or INTEREC. We train Word2Vec for 500 epochs using corpus consisting of items’ textual descriptions with window size is 5, embedding dimension is 100, the number of negative samples is 5.

<sup>6</sup>In our case, we consider *sum* and *mean*. We choose *aggregation* function based on the performance on validation set and report numbers on test set.

Table 8. Retrieval-based Functionally-grounded Interpretability Evaluation with  $k = 10$ , i.e., Query Contains 10 Words

Dataset	Metric	Model				
		CDL	CVAE	JSR	GATE	INTEREC-AE
Cell Phones	HR@20	32.57	23.06	<u>39.50</u>	31.27	<b>42.49</b> <sup>†</sup>
	NDCG@20	11.88	7.49	<u>15.63</u>	11.88	<b>17.04</b> <sup>†</sup>
	HR@50	66.30	52.98	<u>70.83</u>	62.80	<b>72.09</b> <sup>†</sup>
	NDCG@50	18.52	13.36	<u>21.82</u>	18.08	<b>22.89</b> <sup>†</sup>
Toys & Games	HR@20	40.31	26.81	<u>42.37</u>	37.20	<b>46.19</b> <sup>†</sup>
	NDCG@20	17.96	9.28	<u>18.95</u>	15.98	<b>21.79</b> <sup>†</sup>
	HR@50	67.38	55.41	<u>68.74</u>	65.87	<b>71.74</b> <sup>†</sup>
	NDCG@50	23.28	14.90	<u>24.14</u>	21.62	<b>26.82</b> <sup>†</sup>
CDs & Vinyl	HR@20	28.71	29.35	<u>37.66</u>	31.25	<b>47.68</b> <sup>†</sup>
	NDCG@20	10.55	10.82	<u>13.57</u>	10.76	<b>18.10</b> <sup>†</sup>
	HR@50	63.14	63.01	<u>72.28</u>	66.25	<b>78.08</b> <sup>†</sup>
	NDCG@50	16.57	17.41	<u>20.41</u>	17.65	<b>24.13</b> <sup>†</sup>
Citeulike-a	HR@20	71.13	19.27	<u>80.72</u>	53.08	<b>83.05</b> <sup>†</sup>
	NDCG@20	34.11	6.58	<u>42.23</u>	22.83	<b>44.47</b> <sup>†</sup>
	HR@50	91.10	50.35	<u>94.93</u>	82.29	<b>95.52</b> <sup>†</sup>
	NDCG@50	38.11	12.65	<u>45.08</u>	28.63	<b>46.99</b> <sup>†</sup>

The boldfaced and underlined are the highest and the runner-up across models. Statistically significant numbers based on paired t-test are marked by <sup>†</sup> (p-value < 0.05). Unit of reported number are percentage (%).

The results are reported in Table 8. Firstly, INTEREC performs competitively and obtains the best performance on all datasets. This ascertains the role of a dedicated *interpreter* for generating highly relevant words for user preferences interpretation. Additionally, high retrieval performance of our model is attributed to the design of objective function in Section 3.3, i.e., the output of *interpreter* participates in predicting the ground-truth rating of user. As such, the generated words are better at describing user preferences than CDL, CVAE, JSR as these model do not explicitly align generated words with actual user rating. Secondly, among baseline models, JSR performs consistently well, better than CDL and CVAE on all datasets. This suggests that jointly modeling user interactions and textual content is a better choice for interpretability than regularization approach in CDL and CVAE, in which a tradeoff between recommendation and text reconstruction is carefully designed.

Thirdly, the performance of GATE, which is lower than ours, may stem from two reasons, one is noise in text of Amazon datasets and the second is that GATE considers text in sequence, which is very long when concatenating user's adopted item texts. On Citeulike-a dataset, where textual content is less noisy, the performance of GATE is better. We further examine model performance in retrieval task w.r.t. various number of words in query in Table 9. Here, we report performance based on HR@20. For other metrics, we observe the same trend. We have the following observations. First, our proposed model achieves the best performance in retrieval task w.r.t. various number of words in query. Second, increasing the number of words in query generally increases model performance in retrieval task as we use more relevant words to user's preferences. In conclusion, the empirical evidence showcases the power of our *interpreter* in discovering human-comprehensible attributes, i.e., words, to interpret user's interests behind their interactions with items.

We argue that the described *functionally-grounded* evaluation already reflects **fidelity**, a widely used interpretability evaluation [8, 48, 51]. Existing models [8, 48, 51] interpret model prediction so they measure how good *interpreter* approximates black-box model's prediction. Similarly, as



Table 9. Retrieval-based Functionally-grounded Interpretability Evaluation with Varying Number of Words  $k$  in the Query

Dataset	Number of words $k$	Model				
		CDL	CVAE	JSR	GATE	INTEREC-AE
Cell Phones	5	33.79	23.48	<u>38.91</u>	29.62	<b>40.44</b> <sup>†</sup>
	10	32.57	23.06	<u>39.50</u>	31.27	<b>42.49</b> <sup>†</sup>
	15	32.39	23.37	<u>39.74</u>	33.19	<b>42.96</b> <sup>†</sup>
	20	32.17	23.71	<u>39.97</u>	33.96	<b>43.24</b> <sup>†</sup>
Toys & Games	5	39.16	26.73	<u>41.73</u>	34.37	<b>45.20</b> <sup>†</sup>
	10	40.31	26.81	<u>42.37</u>	37.20	<b>46.19</b> <sup>†</sup>
	15	40.39	27.25	<u>42.69</u>	38.44	<b>46.80</b> <sup>†</sup>
	20	40.41	27.30	<u>42.87</u>	39.28	<b>47.20</b> <sup>†</sup>
CDs & Vinyl	5	29.12	29.94	<u>37.18</u>	30.35	<b>45.85</b> <sup>†</sup>
	10	28.71	29.35	<u>37.66</u>	31.25	<b>47.68</b> <sup>†</sup>
	15	28.98	29.30	<u>37.74</u>	33.07	<b>48.69</b> <sup>†</sup>
	20	28.99	29.16	<u>37.79</u>	34.28	<b>49.47</b> <sup>†</sup>
Citeulike-a	5	70.95	19.41	<u>80.17</u>	46.97	<b>80.91</b> <sup>†</sup>
	10	71.13	19.27	<u>80.72</u>	53.08	<b>83.05</b> <sup>†</sup>
	15	71.50	19.91	<u>82.29</u>	56.13	<b>83.80</b> <sup>†</sup>
	20	71.47	20.10	<u>82.45</u>	57.93	<b>83.79</b> <sup>†</sup>

We report HR@20. The same trend applies for other metrics. The boldfaced and underlined are the highest and the runner-up across models. Statistically significant numbers based on paired t-test are marked by <sup>†</sup> (p-value < 0.05). Unit of reported number are percentage (%).

our target is to interpret user’s preferences, we measure the quality of generated words pertaining to capturing user’s preferences through retrieving relevant items for user in a *retrieval* task.

We further examine the effect of architecture design on interpretability in Table 10. The key observations are removing  $\mathbf{V}^{text}$  or *interpreter* results in significantly degraded performance.  $\mathbf{V}^{text}$  encourages  $\mathbf{z}_u$  to capture user’s preferences from textual signals, resulting in better interpretation while training *interpreter* to predict user-item interactions reinforces it to choose words that well capture user interests.

**4.2.3 Qualitative Analysis of Interpretability.** We present a qualitative analysis of INTEREC-AE, GATE and JSR based on these models’ inferred words for user’s preferences interpretation in Table 11. We show two users, namely *1873* and *A1FT98A06ZE4EQ*, and further list the titles of items used in training phase in the first column, as well as the top-10 words generated considered models. Top words with highest  $tf - idf$  score are included for contrasting.

- For these two users, some words produced by GATE are quite general words, e.g., *unfortunately*, *based* or *adds*, which make it difficult to understand user preferences. Contrarily, JSR’s and  $tf - idf$ ’s words are somewhat more relevant.
- In some cases INTEREC-AE identifies some relevant words not discovered by JSR. For example, INTEREC-AE discovers *text* and *mining*, which is one of the interests of the first user (*1873*). For the second user (*A1FT98A06ZE4EQ*), it seems that she bought items to protect her phone. INTEREC discovers related words, e.g., *screen* and *protectors*.
- INTEREC and JSR are more generalizable than  $tf - idf$  by the ability to discovering preference words beyond user’s adopted items’ texts, e.g., *classification* or *matte*.

We further show the generated words as word clouds in Figure 2 and 3. In these figures, the bigger a word is, the higher its predicted score by INTEREC-AE is. Position and color are set

Table 10. Retrieval-based Functionally-grounded Interpretability Evaluation when Removing *interpreter* and Removing  $V^{text}$ 

Dataset	Metric	INTEREC-AE		
		full model	without <i>interpreter</i>	without $V^{text}$
Cell Phones	HR@20	<b>42.49</b>	37.70	18.72
	NDCG@20	<b>17.04</b>	14.80	6.41
	HR@50	<b>72.09</b>	67.19	48.55
	NDCG@50	<b>22.89</b>	20.61	12.23
Toys & Games	HR@20	<b>46.19</b>	43.95	19.25
	NDCG@20	<b>21.79</b>	20.00	6.69
	HR@50	<b>71.74</b>	70.09	49.12
	NDCG@50	<b>26.82</b>	25.15	12.52
CDs & Vinyl	HR@20	<b>47.68</b>	41.32	16.96
	NDCG@20	<b>18.10</b>	14.91	5.58
	HR@50	<b>78.08</b>	73.71	50.16
	NDCG@50	<b>24.13</b>	21.32	12.05
Citeulike-a	HR@20	<b>83.05</b>	80.81	18.78
	NDCG@20	<b>44.47</b>	41.45	6.56
	HR@50	<b>95.52</b>	94.70	48.12
	NDCG@50	<b>46.99</b>	44.25	12.28

Unit of reported number is percentage (%).

Table 11. Examples of Inferred Words for user in Citeulike-a Dataset (ID: 1873) in the First Row and user (ID:A1FT98A06ZE4EQ) in Cell Phones Dataset in the Second Row

Titles of Adopted Items	INTEREC	JSR	GATE	Tf-idf
1. A Brief Survey of Web Data Extraction Tools	learning	learning	calculus	extraction
2. A Tutorial on Support Vector Machines for Pattern Recognition	machine	web	good	ie
3. Adaptive information extraction	mining	semantic	effort	web
4. Automatic web news extraction using tree edit distance	web	training	shallow	data
5. A Survey of Web Information Extraction Systems	extraction	machine	hope	learning
6. Relational Learning of Pattern - Match Rules for Information Extraction	text	extraction	structures	dimension
7. BoosTexter: A Boosting - based System for Text Categorization	training	task	correctly	text
8. Pattern Recognition and Machine Learning (Information Science and Statistics)	<u>classification</u>	search	article	machine
	semantic	tasks	unfortunately	categorization
	recognition	<u>classification</u>	based	pattern
1. Generiks TM iPhone 4&4S ANTI - FINGERPRINT/ANTI - GLARE Screen Protectors	apple	iphone	adds	iphone
2. Generiks TM iPhone 4 / 4S *CLEAR* Screen Protectors	iphone	amp	trademarks	pink
3. Snap - on Rubber Coated Case for Apple iPhone 4 4S 4GS 4G AT&T / Verizon, Pink / Black	kitty	apple	protected	protectors
4. Deluxe AT&T Verizon White For Iphone 4 4S 4G Case Cover with Kickstand	anti	pink	phone	cover
5. 3d Hello Kitty Pink Ribbon Case / cover / protector Fits All Models of Iphone 4 & 4s	glare	hot	easy	name
6. Leegoal Lightweight Hybrid Bumper Skin Back Case Cover for iPhone 5 5G Pink	pink	verizon	keeps	deluxe
	hello	white	accessory	verizon
	screen	sprint	endorsed	at&t
	protectors	back	controls	g
	<u>matte</u>	stand	iphone	ribbon

Underlined words are outside user's adopted items' texts.

randomly. It is clear that human can easily understand user's interest topic in each word cloud. This analysis, based on a few examples, is not meant to be a formal comparison per se. Rather, it helps to illustrate some of the qualitative differences that underlie the quantitative comparison of interpretability presented in the previous tables.

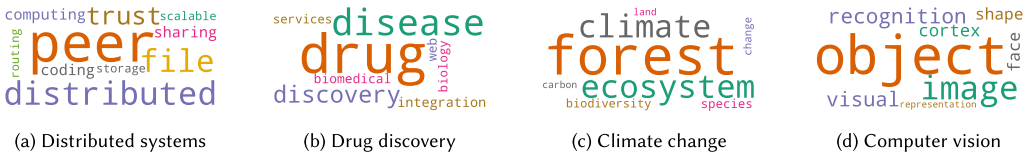


Fig. 2. Inferred words for users on Citeulike-a dataset. Each word cloud represents topic of interest for one user. Best viewed in color.

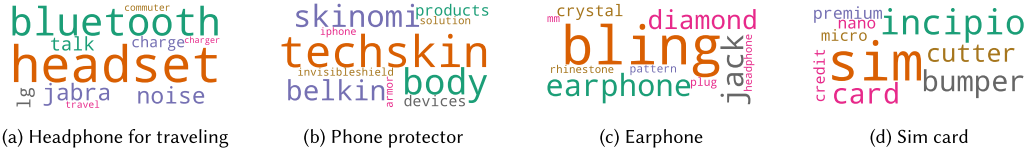


Fig. 3. Inferred words for users on Cell Phones dataset. Each word cloud represents topic of interest for one user. Best viewed in color.

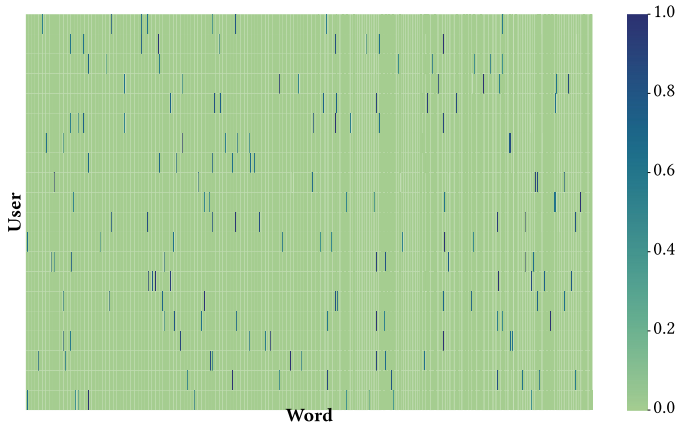


Fig. 4. Visualization of activated words produced by our proposed model on Cell Phone dataset. Each row represents a randomly selected user. In each row, we visualize top 10 words with highest activated scores of a user. Scores are normalized between 0 and 1 to ease the visualization. Each column represents a word in vocabulary. It is clear that each user has her own set of top activated words, showing that our proposed model produces diverse interpretation of user’s preferences.

4.2.4 *Diversity of Interpretability.* To verify that our proposed approach achieves diversity of interpretability, we visualize the activated word scores of randomly selected users in Figure 4. We observe that each user has her own activated word score pattern. Concretely, for each user, top words with highest score are different from a user to one another. This ascertains our model’s ability to produce diverse set of words to interpret user’s preferences.

### 4.3 Architecture and Hyper-Parameter Analysis

We investigate the impacts of architecture and hyper-parameters on recommendation and interpretability objectives of INTEREC-AE, which achieves competitive performance on four chosen datasets. Table 12 reports recommendation accuracy and functionally-grounded evaluation of interpretability.

Table 12. Results in our Ablation Analysis

Dataset	Metric	Interpretability					Recommendation				
		(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
Cell Phones	HR@20	42.49	39.89	23.81	42.56	30.57	10.57	10.39	10.66	10.40	10.33
	NDCG@20	17.04	15.84	8.66	16.99	11.61	4.46	4.45	4.52	4.44	4.43
	HR@50	72.09	69.36	54.07	72.03	61.71	17.78	17.68	17.96	17.53	17.52
	NDCG@50	22.89	21.65	14.58	22.82	17.73	5.88	5.89	5.96	5.84	5.85
Toys & Games	HR@20	46.19	44.67	27.93	46.36	31.21	9.35	9.09	9.24	8.76	8.38
	NDCG@20	21.79	20.56	10.69	21.69	12.66	3.85	3.78	3.84	3.70	3.56
	HR@50	71.74	70.50	58.43	71.61	60.70	15.39	15.23	15.05	14.69	13.97
	NDCG@50	26.82	25.64	16.67	26.67	18.44	5.05	4.99	4.99	4.88	4.67
CDs & Vinyl	HR@20	47.68	43.55	17.64	47.33	19.77	9.36	9.15	9.33	9.33	9.26
	NDCG@20	18.10	15.95	5.76	17.96	6.44	3.97	3.86	3.97	3.97	3.95
	HR@50	78.08	75.33	51.47	78.00	54.50	15.27	15.11	15.36	15.24	15.20
	NDCG@50	24.13	22.25	12.35	24.05	13.22	5.14	5.03	5.17	5.13	5.12
Citeulike-a	HR@20	83.05	82.40	36.28	82.57	47.86	24.94	23.75	24.93	24.96	25.17
	NDCG@20	44.47	43.20	15.05	44.10	21.92	11.29	10.67	11.30	11.41	11.40
	HR@50	95.52	95.34	65.83	95.49	74.49	38.10	36.76	38.07	37.56	37.63
	NDCG@50	46.99	45.80	20.86	46.71	27.18	13.89	13.24	13.91	13.90	13.87

Each column from (2) - (5) is a variant of INTEREC-AE. (1) INTEREC-AE. (2): *sigmoid* (in Equation (6)) is replaced by standard *softmax*. (3): fine tuning  $\mathbf{K}$ . (4): fine tuning  $\mathbf{V}^{text}$ . (5): fine tuning both  $\mathbf{K}$  and  $\mathbf{V}^{text}$  (fine tuning mean that we allow update  $\mathbf{K}$  and  $\mathbf{V}^{text}$  in the second stage in Algorithm 1). Unit of reported number is percentage (%).

**Sigmoid vs. Softmax** in Equation (6). Column (2) in Table 12 shows that if we use *softmax* in place of *sigmoid* (Equation (6)), on *retrieval* we observe a 1%–3% drop on Amazon subsets, larger than those on Citeulike-a. Similarly for *recommendation*. Empirically *interpreter* equipped with *sigmoid* is more effective than standard *softmax* for our case.

**Effects of  $\epsilon$ .** Recall that  $\epsilon$  controls the *conciseness* of interpretability. We vary  $\epsilon$  in Equation (8) and report model performance in Figure 5. Generally, we observe that for *recommendation* objective, there is a consistency between 4 datasets that the optimal value of  $\epsilon$  falls in range [0.5 – 0.7], which supports our claim that  $\epsilon$  is between 0 and 1. Regarding *interpretability* objective, on Cell Phones and CDs & Vinyl dataset, the value of  $\epsilon$  is consistent with the one in *recommendation*. On Toys & Games and Citeulike-a dataset, *retrieval* performance peaks when the value of  $\epsilon$  is around 0.1 and 0.2, respectively. Overall, the results support our hypothesis in Section 3.2 that  $\epsilon$  is between 0 and 1.

**Fixing vs. Updating  $\mathbf{K}$  and  $\mathbf{V}^{text}$  in the second stage training in Algorithm 1.** We first fine-tune  $\mathbf{K}$  and keep others as default choice. Next, we freeze  $\mathbf{K}$  and fine-tune  $\mathbf{V}^{text}$ . Finally, we fine-tune both  $\mathbf{K}$  and  $\mathbf{V}^{text}$ . The results of these experiments are shown on columns (3)–(5) on Table 12. Generally, fine-tuning one or both of the embeddings results in degraded performance. We intuit that these embeddings bring with them useful independent textual signals that would be overridden by collaborative filtering signals if floated.

As analyzed in previous sections, the performance of INTEREC is based on that of *recommender*. Therefore, in future application of our proposed architecture, one should pay attention and carefully choose the values for above mentioned designs and hyper-parameter choices w.r.t. the choice of *recommender*.

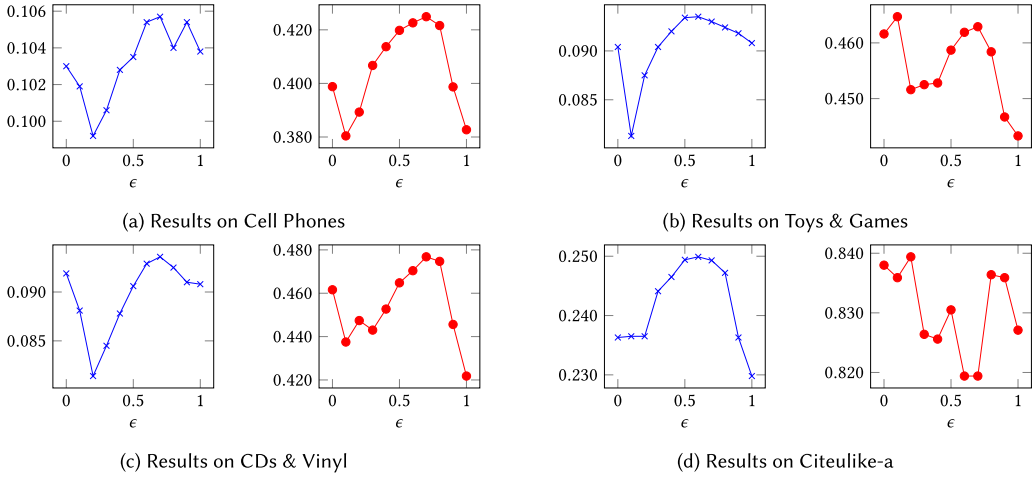


Fig. 5. Results on 4 datasets with various values of  $\epsilon$ . Blue lines represent HR@20 in recommendation task while red lines represent HR@20 on retrieval-based functional-grounded evaluation of interpretability.

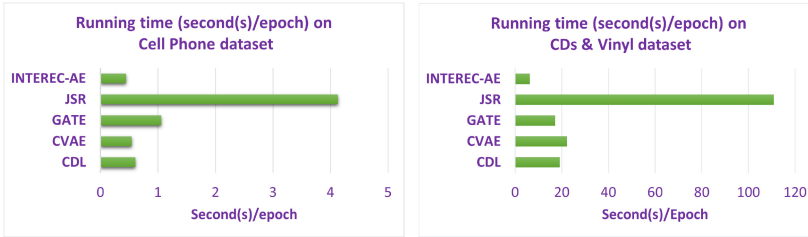


Fig. 6. Runtime analysis (seconds/epoch) of all models on Cell Phone dataset and CDs & Vinyl dataset. All models are trained with batch size 256. Reported numbers are averaged over 10 runs.

**Run Time Analysis.** We briefly discuss the run time of INTEREC-AE vis-a-vis the user-oriented textual-aware baselines. Figure 6 reports the number of seconds taken to run a single epoch in the examined models. Firstly, on small dataset, Cell Phones, the running time of INTEREC-AE is close to those of CDL and CVAE and nearly half of GATE. Recall that GATE requires extra time to process neighbor information. Secondly, on large dataset, CDs & Vinyl, INTEREC-AE maintains its time efficiency while an extra time is required in CDL and CVAE since these models store explicitly a huge number of user representations. GATE performs similarly to CDL and faster than CVAE. Lastly, on both dataset, JSR is not time-efficient due to negative sampling, which increases the number of ratings and the running time.

## 5 CONCLUSION

We propose INTEREC, a novel unified architecture for joint learning a neural *recommender* and an *interpreter*. Our work adopts a new angle to existing content-aware recommendation models by employing textual content for user’s preferences interpretation besides sparsity alleviation. In particular, our model provides *local interpretability* of user’s preferences underlying her adoptions in terms of human comprehensible attributes described by natural language words. The means of doing so is a dedicated *interpreter* which relies on user representation from *recommender*. A key-value memory network is used to implement *interpreter*, leading to a generalized user representation by discovering words going beyond user’s interacted items’ contents.

There are several research directions for future work to further build upon INTEREC. The first one is the investigation of the proposed architecture considering multi-interest recommender, which represents a user by multiple embedding vectors. Second, organizing textual content into a structure, e.g., topic modeling, and leveraging structured units for interpretation of user's preferences. Last but not least, other type of side information, e.g., knowledge graph, could be worth exploring to gain better insights into user's preferences underlying their item adoptions.

## REFERENCES

- [1] Tameem Adel, Zoubin Ghahramani, and Adrian Weller. 2018. Discovering interpretable representations for both deep generative and discriminative models. In *Proceedings of the 35th International Conference on Machine Learning*. 50–59.
- [2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *the Journal of Machine Learning Research* 3 (2003), 993–1022.
- [3] Desheng Cai, Shengsheng Qian, Quan Fang, Jun Hu, Wenkui Ding, and Changsheng Xu. 2023. Heterogeneous graph contrastive learning network for personalized micro-video recommendation. *IEEE Transactions on Multimedia* 25 (2023), 2761–2773.
- [4] Desheng Cai, Shengsheng Qian, Quan Fang, Jun Hu, and Changsheng Xu. 2023. User cold-start recommendation via inductive heterogeneous graph neural network. *ACM Transactions on Information System* 41, 3 (2023), 27 pages.
- [5] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. 2019. This looks like that: Deep learning for interpretable image recognition. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*.
- [6] Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An efficient adaptive transfer neural network for social-aware recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 225–234.
- [7] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient neural matrix factorization without sampling for recommendation. *ACM Transactions on Information System* 38, 2 (2020), 28 pages.
- [8] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *Proceedings of the 35th International Conference on Machine Learning*. 883–892.
- [9] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 335–344.
- [10] Mengru Chen, Chao Huang, Lianghao Xia, Wei Wei, Yong Xu, and Ronghua Luo. 2023. Heterogeneous graph contrastive learning for recommendation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*. 544–552.
- [11] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiayi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. 108–116.
- [12] Zhi Chen, Yijie Bei, and Cynthia Rudin. 2020. Concept whitening for interpretable image recognition. *Nature Machine Intelligence* 2, 12 (2020), 772–782.
- [13] Finale Doshi-Velez and Been Kim. 2017. *Towards A Rigorous Science of Interpretable Machine Learning*. arXiv:1702.08608 [stat.ML].
- [14] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *Proceedings of the World Wide Web Conference*. 417–426.
- [15] Li Gao, Jia Wu, Chuan Zhou, and Yue Hu. 2017. Collaborative dynamic sparse topic regression with user profile evolution for item recommendation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)*. AAAI Press, 1316–1322.
- [16] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, and Dawei Yin. 2020. Hierarchical user profiling for e-commerce recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 223–231.
- [17] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*. 507–517.
- [18] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 639–648.
- [19] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. NAIS: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 30, 12 (2018), 2354–2366.



- [20] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182.
- [21] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. *spaCy: Industrial-strength Natural Language Processing in Python*. DOI: <https://doi.org/10.5281/zenodo.1212303>
- [22] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 2018. Leveraging meta-path based context for top- N recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1531–1540.
- [23] Liang Hu, Songlei Jian, Longbing Cao, and Qingkui Chen. 2018. Interpretable recommendation via attraction modeling: Learning multilevel attractiveness over multimodal movie contents. In *Proceedings of the IJCAI International Joint Conference on Artificial Intelligence*. 3400–3406.
- [24] Liang Hu, Songlei Jian, Longbing Cao, Zhiping Gu, Qingkui Chen, and Artak Amirbekyan. 2019. HERS: Modeling influential contexts with heterogeneous relations for sparse and cold-start recommendation. *AAAI* 33, 1 (2019), 3830–3837.
- [25] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 8th IEEE International Conference on Data Mining*. 263–272.
- [26] Dmitry Kazhdan, Boty Dimanov, Mateja Jamnik, Pietro Liò, and Adrian Weller. 2020. Now you see me (CME): Concept-based model extraction. In *Proceedings of the CIKM 2020 Workshops*.
- [27] Tom Kenter and Maarten de Rijke. 2015. Short text similarity with word embeddings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1411–1420.
- [28] Dong Hyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 233–240.
- [29] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.
- [30] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*. 5338–5348.
- [31] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. 2018. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.
- [32] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 305–314.
- [33] Dawen Liang, Rahul G. Krishnan, Mathew D. Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*. 689–698.
- [34] Shangsong Liang. 2019. Collaborative, dynamic and diversified user profiling. *AAAI* 33, 1 (2019), 4269–4276.
- [35] Shangsong Liang, Xiangliang Zhang, Zhaochun Ren, and Evangelos Kanoulas. 2018. Dynamic embeddings for user profiling in twitter. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1764–1773.
- [36] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 2181–2187.
- [37] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2019. Explainable outfit recommendation with joint outfit matching and comment generation. *IEEE TKDE* 32, 8 (2019), 1502–1516.
- [38] Zachary C. Lipton. 2017. *The Mythos of Model Interpretability*. arXiv:1606.03490.
- [39] Yang Liu, Liang Chen, Xiangnan He, Jiaying Peng, Zibin Zheng, and Jie Tang. 2022. Modelling high-order social relations for item recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 9 (2022), 4385–4397.
- [40] Zhongqi Lu, Sinno Jialin Pan, Yong Li, Jie Jiang, and Qiang Yang. 2016. Collaborative evolution for user profiling in recommender systems. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. 3804–3810.
- [41] Yupeng Luo, Shangsong Liang, and Zaiqiao Meng. 2019. Constrained co-embedding model for user profiling in question answering communities. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 439–448.
- [42] Chen Ma, Peng Kang, Bin Wu, Qinglong Wang, and Xue Liu. 2019. Gated attentive-autoencoder for content-aware recommendation. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 519–527.
- [43] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly learning explainable rules for recommendation with knowledge graph. In *Proceedings of the World Wide Web Conference*. 1210–1221.
- [44] Julian McAuley, Christopher Targe, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR Conference on Research and Development in Information Retrieval*. 43–52.

- [45] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Advances in Neural Information Processing Systems*. 3111–3119.
- [46] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. *Key-Value Memory Networks for Directly Reading Documents*. arXiv:1606.03126.
- [47] Deng Pan, Xiangrui Li, Xin Li, and Dongxiao Zhu. 2020. Explainable recommendation via interpretable feature mapping and evaluation of explainability. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI'20)*. 2690–2696.
- [48] Jayneel Parekh, Pavlo Mozharovskiy, and Florence d'Alché-Buc. 2021. A framework to learn with interpretation. In *Proceedings of the Advances in Neural Information Processing Systems*. 24273–24285.
- [49] Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. 45–50.
- [50] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. 2017. Social collaborative viewpoint regression with explainable recommendations. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*. 485–494.
- [51] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1135–1144.
- [52] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215.
- [53] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*. 111–112.
- [54] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*. 618–626.
- [55] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the 11th ACM Conference on Recommender Systems*. 297–305.
- [56] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Proceedings of the Advances in Neural Information Processing Systems*. 2440–2448.
- [57] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference*. 729–739.
- [58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*. 6000–6010.
- [59] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a Local Denoising Criterion. *Journal of Machine Learning Research* 11, 110 (2010), 3371–3408.
- [60] Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 448–456.
- [61] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards representation alignment and uniformity in collaborative filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1816–1825.
- [62] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1235–1244.
- [63] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 968–977.
- [64] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable recommendation via multi-task learning in opinionated text data. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. 165–174.
- [65] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 950–958.
- [66] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 165–174.

- [67] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *AAAI*.
- [68] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 235–244.
- [69] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*. 153–162.
- [70] Wenyi Xiao, Huan Zhao, Haojie Pan, Yangqiu Song, Vincent W. Zheng, and Qiang Yang. 2019. Beyond personalization: Social content recommendation for creator equality and consumer satisfaction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 235–245.
- [71] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon Jose. 2019. Relational collaborative filtering: Modeling multiple item relations for recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 125–134.
- [72] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. 2022. Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering* 34, 10 (2022), 4854–4873.
- [73] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. 2019. INVASE: Instance-wise variable selection using neural networks. In *Proceedings of the International Conference on Learning Representations*.
- [74] Hamed Zamani and W. Bruce Croft. 2020. Learning a joint search and recommendation model from user-item interactions. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 717–725.
- [75] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *ECCV*.
- [76] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 353–362.
- [77] Yongfeng Zhang and Xu Chen. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends in Information Retrieval* 14, 1 (2020), 1–101.
- [78] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 83–92.

Received 6 April 2023; revised 27 August 2023; accepted 1 September 2023