A Capsule Network for Recommendation and Explaining What You Like and Dislike

Chenliang Li[1], Cong Quan[2], Li Peng[3], Yunwei Qi[3], Yuming Deng[3], Libing Wu[2]

1. Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, China

cllee@whu.edu.cn

2. School of Computer Science, Wuhan University, Wuhan, 430072, China

{quancong,wu}@whu.edu.cn

3. Alibaba Group, Hangzhou, China

{muchen.pl,yunwei.qyw,yuming.dym}@alibaba-inc.com

ABSTRACT

User reviews contain rich semantics towards the preference of users to features of items. Recently, many deep learning based solutions have been proposed by exploiting reviews for recommendation. The attention mechanism is mainly adopted in these works to identify words or aspects that are important for rating prediction. However, it is still hard to understand whether a user likes or dislikes an aspect of an item according to what viewpoint the user holds and to what extent, without examining the review details. Here, we consider a pair of *a viewpoint held by a user* and *an aspect of an item* as a ***logic unit***. Reasoning a rating behavior by discovering the informative logic units from the reviews and resolving their corresponding sentiments could enable a better rating prediction with explanation.

To this end, in this paper, we propose a **ca**psule network based model for **r**ating **p**rediction with user reviews, named CARP. For each user-item pair, CARP is devised to extract the informative logic units from the reviews and infer their corresponding sentiments. The model firstly extracts the viewpoints and aspects from the user and item review documents respectively. Then we derive the representation of each logic unit based on its constituent viewpoint and aspect. A sentiment capsule architecture with a novel *Routing by Bi-Agreement* mechanism is proposed to identify the informative logic unit and the sentiment based representations in user-item level for rating prediction. Extensive experiments are conducted over seven real-world datasets with diverse characteristics. Our results demonstrate that the proposed CARP obtains substantial performance gain over recently proposed state-of-the-art models in terms of prediction accuracy. Further analysis shows that our model can successfully discover the interpretable reasons at a finer level of granularity.

KEYWORDS

Recommender System, User Reviews, Deep Learning

Figure 1: Review examples for *Apple iPhone X* in Amazon.

1 INTRODUCTION

Many E-Commerce platforms allow a user to share her experience of a purchased item in the form of review, along with a numerical rating, such as Yelp and Amazon (ref. Figure 1). These textual reviews have played an increasingly important role for E-Commerce platforms in supporting personalized recommendation. Many recommender systems have been developed by exploiting the semantic information covered in reviews [1, 3–6, 15, 18, 20, 24, 26, 27, 29, 37, 39] against using naturally sparse user-item ratings, leading to signficant performance gain.

The earlier solutions choose to adopt topic modeling or nonnegative matrix factorization over reviews and represent a user/item semantically [1, 18, 20, 26, 29]. This modeling paradigm is recently suppressed by the revival of deep learning techniques. Specifically, empowered by continuous real-valued vector representations and semantic composition over contextual information, uptodate review-based deep learning models significantly push the frontier of the state-of-the-art further. Examples include DeepCoNN [39], D-Attn [24], TransNet [3], ANR [7] and MPCN [27].

Though significant performance gain is achieved by these efforts, these models do not explain what really happens in a user's mind when making a rating. It is reasonable to assume that a user would attach different importance to different aspects of an item. Similarly, different users would care a specific aspect of an item to different

extent [6]. When we further consider the sentiment information, the things become more interesting. A user could hold opposite opinions towards an item by considering different aspects. Hence, the whole rating score can be considered as an indication of an item's relative merit (*i.e.,* a compromise on the virtues and achilles heels). The examples demonstrated in Figure 1 fall in this line. Here, we refer to all reviews wirtten by a user as *user document*. A user document mainly contains her personal viewpoints towards different items. For example, a user could prefer *outdoor activities* and therefore appreciate the related aspects (*e.g., low profile, easy storage*) for an item. Also, an *item document* can be formed by merging all reviews written for the item. An item document can be considered as a summary of the item's various aspects, where the important aspects can be easily exposed [19, 26]. Here, to reason a rating behavior, we wish to understand *whether a user likes or dislikes an aspect of an item according to what viewpoint she holds, and to what extent.*

To this end, in this paper, we propose a **ca**psule network based model for **r**ating **p**rediction with user reviews, named CARP. In detail, CARP consists of two components: *viewpoint and aspect extraction* and *sentiment capsules*. Firstly, a variant of self-attention stacked over a convolutional layer is adopted to extract the viewpoints and aspects mentioned from the user and item documents respectively. Given a user-item pair, we pack a user viewpoint and an item aspect together to form a *logic unit*. This notion serves as a proxy for causes behind a rating behavior. However, not all logic units formed by this simple pairing would make sense in real-world. We refer to the semantically plausible logic units as *informative logic units*. For example, a viewpoint about *outdoor enthusiast* and an item aspect about *easy storage* would form an informative logic unit. On the other hand, the pair of the viewpoint about *elegant appearance* and the same aspect about *easy storage* would just result in a horrible haze.

To identify the informative logic unit, we derive the representations of all possible logic units and feed them into the sentiment capsules. Specifically, we utilize a positive capsule and a negative capsule to represent the user's attitudes towards the item. By devising a new iterative routing mechanism - *Routing by Bi-Agreement* - we enable the two sentiment capsules to jointly identify the informative logic units with positive and negative sentiments held by the user respectively. The output vectors produced by these two sentiment capsules encode to what extent the user likes or dislikes the item respectively. Meanwhile, the vector lengths suggest the probability of each of these two sentiments. For a given user-item pair, the overall rating is then estimated based on these magnitudes and odds in two sentiment poles. At last, we need to emphasize that CARP works in a review-driven and end-to-end fashion. To guarantee the proper sentiment modeling, we cast CARP as a multi-task learning process such that implicit sentiment information provided by a rating score is exploited. Specifically, we introduce a sentiment matching process by classifying the sentiment of a review according to its rating score. Note that no human labeling or external NLP tool is required to aid the training of CARP.

Our work differs fundamentally from the prior works, since we aim to make a bridge connecting the reviews and rating scores with causes and effects, in terms of user viewpoints, item aspects and sentiments. Back to existing deep learning based solutions, only

important words or aspects in the reviews are highlighted by the representation learning process [7, 19, 24, 27]. It is difficult to infer what rules a user applies in her mind and the effects separately. Overall, our key contributions are summarized as below:

- We propose a novel deep learning model that exploits reviews for rating prediction and explanation. This is the very first attempt to explicitly model the reasoning process underlying a rating behavior in E-Commerce, by jointly considering user viewpoints, item aspects and sentiments.
- We introduce a capsule-based architecture to jointly identify informative logic units and estimate their effects in terms of sentiment analysis. We further propose Routing by Bi-Agreement, a new iterative dynamic routing process for capsule networks.
- To enable sentiment modeling without using any external resource, a multi-task learning process is proposed for model optimization, which can successfully exploit the implicit sentiment information provided by a rating score.
- On seven real-world datasets with diverse characteristics, our results demonstrate the superiority of the proposed CARP in terms of rating prediction accuracy, performance robustness and explanation at fine-grained level.

2 RELATED WORK

In this section, we briefly review three different areas which are highly relevant to our work.

2.1 Review-based Recommender Systems

Recent years, capitalizing user reviews to enhance the precision and the interpretability of recommendation have been investigated and verified by many works [1, 3, 4, 6, 8, 15, 18, 20, 24, 26, 27, 29, 37, 39]. In earlier days, many efforts are made to extract semantic features from reviews with the topic modeling techniques [2, 14]. These works integrate the latent semantic topics into the factor learning models [1, 18, 20, 26, 29]. TLFM proposes two separate factor learning models to exploit both sentiment-consistency and text-consistency of users and items [25]. Then they unify these two views together to conduct rating prediction. However, the proposed solution requires the review of the target user-item pair as input, which is unrealistic and thus turns the recommendation task into a sentiment analysis task. CDL [30] proposes to couple SADE over reviews and PMF [23]. Since the bag-of-word representation schema is used in these works, signficant information loss is expected due to the missing of contextual information.

Recently, many works are proposed to model contextual information from reviews for better recommendation with deep learning techniques [3, 4, 24, 27, 39]. Convolutional neural networks (CNN) [16] and Recurrent Neural Network (RNN) [9, 21] are mainly used to composite the semantic contextual information into the continuous real-valued vector representation. ConvMF shares the same architecture with CDL and uses CNN to extract item characteristics from the item description [15]. TARMF uses the features extracted from both user document and item document to calibrate the latent factors in the factor learning paradigm [19]. DeepCoNN uses parallel CNN networks to uncover semantic features from user and item documents [39]. The extracted features are then fed

into the Factorization Machine for rating prediction. TransNet augments DeepCoNN with an additional transform layer to infer the representation of the target user-item review that is unavailable during the rating prediction [3]. D-Attn leverages global and local attention mechanisms to identify important words in the review documents for rating prediction [24]. Note that each review contains different semantic information. NARRE argues that reviews should be treated differently upon different user-item pairs and proposes to employ an attention mechanism to select useful reviews for rating prediction [4]. Likewise, MPCN exploits a pointer-based co-attention schema to enable a multi-hierarchical information selection. Both important reviews and their important words can be identified towards better rating prediction in MPCN [27].

2.2 Aspect-based Recommender Systems

Many efforts are devoted to model user opinions from reviews for transparent recommendation, namely aspect-based recommender systems, which can be further divided into two categories. The first category resorts to using the external NLP tools to extract aspects and sentiments from reviews. For example, EFM and MTER generate a sentiment lexicon through a phrase-level NLP tool [31, 38]. Similarly, TriRank utilizes the extracted aspects to construct a tripartite graph over users and items [11]. These works heavily rely on the performance of the external toolkit.

The second category aims to automatically infer explainable aspects from reviews by devising an internal component. JMARS learns the aspect representations of user and item by using topic modeling to conduct collaborative filtering [8]. AFLM proposes an aspect-aware topic model over the reviews to learn diferent aspects of user and item in topic space [6]. Very recently, ANR proposes a co-attention mechanism in the neural architecture to automatically estimate aspect-level ratings and aspect-level importance in an end-to-end fashion [7].

Many of the above review-based and aspect-based solutions aim to identify the important words or aspects to enhance the recommendation and facilitate explanation. However, only an incomplete picture is drawn by them towards the complex reasoning process underlying a rating behavior. They are incapable of telling whether a user likes or dislikes an aspect of an item according to what viewpoint the user holds and to what extent. In this work, the proposed CARP can be seen as a preliminary step to fill in the missing part.

2.3 Capsule Networks for NLP

The capsule network is proposed as a hierarchical architecture to model the complex relations among latent features [13]. The affiliated dynamic routing (Routing by Agreement) mechanism in a capsule ensures that low-level features can be selectively aggregated to form high-level features [22]. Recently, this notion has been applied to some NLP tasks, including relation extraction [36], text classification [35], zero-shot user intent detection [33] and multi-task learning [34]. In this work, we exploit the capsule-based architecture with a new proposed Routing by Bi-Agreement (RBiA) mechanism to achieve multiple objectives jointly. RBiA calculates the inter-capsule and intra-capsule agreements to derive the output of a capsule, which would be a useful complement to the existing studies.

3 THE PROPOSED MODEL

In this section, we present the proposed capsule network CARP and Figure 2 illustrates its overall architecture.

3.1 Viewpoint and Aspect Extraction

As previously mentioned, a user document D_u is formed by merging all the reviews written by user u. And an item document D_i can be formed in a similar way. We firstly extract the user veiwpoints and item aspects from the correponding documents respectively. Here, we give a detailed description about viewpoint extraction in CARP. The same procedure applies for aspect extraction.

Context Encoding. Given a user document $D_u = (w_1, w_2, ..., w_l)$ where l is the document length in number of words, we first project each word to its embedding representation: $\mathbf{D}_u = (\mathbf{e}_1, ..., \mathbf{e}_l)$. Here, $\mathbf{e}_j \in R^d$ is the embedding vector for word at the j-th position, and d is the embedding size. Then, to extract possible aspect-specific information around each word, we perform convolution operation with ReLU (*i.e.,* Rectified Linear Unit) as the activation function. Specifically, the context is encoded by using a window spanning $\frac{c-1}{2}$ words on both side of each word. The resultant latent feature vectors are $[\mathbf{c}_1, ..., \mathbf{c}_l]$ and $\mathbf{c}_j \in R^n$ is the latent contextual feature vector for j-th word, and n is the number of filters.

Self-Attention. Intuitively, not all words in the document are important for each viewpoint. To identify which features carried by $\mathbf{c}_j$ are relevant to each viewpoint, we leverage a viewpoint-specific gating mechanism:

$$\mathbf{s}_{u,x,j} = \mathbf{c}_j \odot \sigma(\mathbf{W}_{x,1}\mathbf{c}_j + \mathbf{W}_{x,2}\mathbf{q}_{u,x} + \mathbf{b}_x) \tag{1}$$

where $\mathbf{W}_{x,1}, \mathbf{W}_{x,2} \in R^{n \times n}$ and $\mathbf{b}_x \in R^n$ are transform matrices and bias vector respectively for the x-th viewpoint, σ is the sigmoid activation and $\odot$ is the element-wise product operation, $\mathbf{q}_{u,x}$ is the embedding of x-th viewpoint shared for all users, which is learned by model optimization. We then extract the contextual viewpoint representation $\mathbf{p}_{u,x,j}$ from $\mathbf{s}_{u,x,j}$ through a projection: $\mathbf{p}_{u,x,j} = \mathbf{W}_p\mathbf{s}_{u,x,j}$ where $\mathbf{W}_p \in R^{k \times n}$ is a viewpoint-shared transform matrix. It is reasonable that the viewpoint-specific semantic of the same word within similar contexts could be different. For example, opposite sentiments are held by same word 'long' in contexts "the battery can sustains a *long* time" and "the waiting time is *long*". The aspect-specific gating and projection help us extract the viewpoint relevant information precisely by disambiguating the semantics of each word.

It is natural that more viewpoint-specific words mentioned in the reviews, more firmly the viewpoint held by the user. For example, 'performance', 'durable', and 'practical' would be frequently mentioned by a pragmatist while 'lovely' and 'aesthetics' are supposed to dominate the reviews written by appeareance-valuing users. In CARP, we propose to leverage the self-attention mechanism to derive the user viewpoint from her document. Specifically, We first derive the rudimentary representation of the viewpoint by taking the average sum of the constituent viewpoint-specific contextual embeddings in user document (*i.e.,* $\mathbf{v}_{u,x} = \frac{1}{l}\sum_j \mathbf{p}_{u,x,j}$). Then the intra-attention weights are calculated as $attn_{u,x,j} = softmax(\mathbf{p}_{u,x,j}^\top \mathbf{v}_{u,x})$. At last, the viewpoint $\mathbf{v}_{u,x}$ is represented
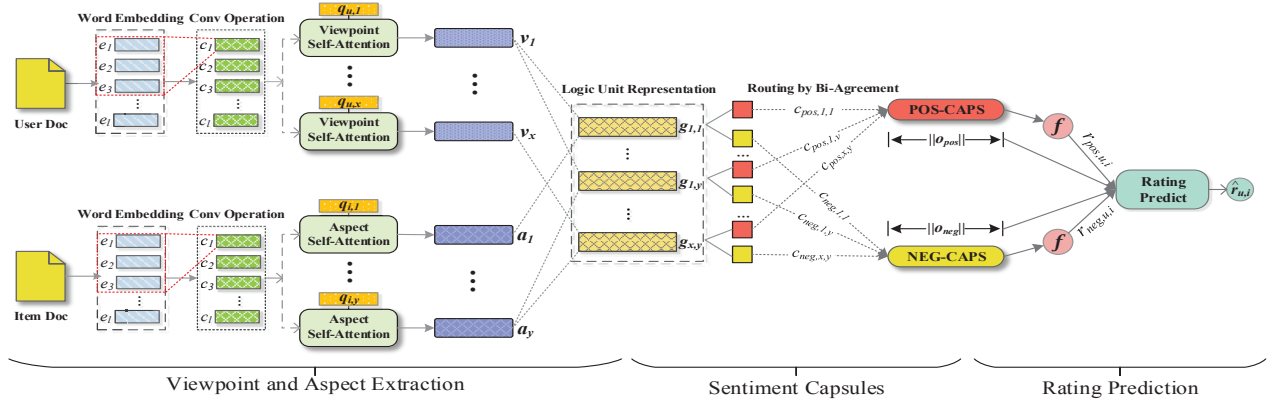
Figure 2: The network architecture of CARP.

as a weighted sum:

$$\mathbf{v}_{u,x} = \sum_j attn_{u,x,j} \mathbf{p}_{u,x,j} \tag{2}$$

The intra-attention mechanism enables the viewpoint extraction to capture the features which are consistently important for different viewpoints. For model simplicity, we restrict the viewpoint number of a user and aspect number of an item to be the same. Following the same procedure, we extract M aspects for an item from the corresponding item document with a different set of parameters (*e.g.,* the convolution operation, aspect embedding and gating mechanism).
Logic Unit Representation. With the user viewpoints and item aspects extracted above, we need to identify the rules that a user applies when making a rating towards a specific item. Actually, for decision making, these rules take the form of various causes and their effects. It is reasonable that the user and the item should appear on both sides of a cause. Hence, a cause is formed by packing a user viewpoint with an item aspect together, which is named a *logic unit*. Specifically, given the x-th viewpoint $\mathbf{v}_{u,x}$ of user u and y-th aspect $\mathbf{a}_{i,y}$ of item i, the representation $\mathbf{g}_{x,y}$ of the corresponding logic unit $g_{x,y}$ is derived as follows:

$$\mathbf{g}_{x,y} = [(\mathbf{v}_{u,x} - \mathbf{a}_{i,y}) \oplus (\mathbf{v}_{u,x} \odot \mathbf{a}_{i,y})] \tag{3}$$

where $\oplus$ is the vector concatenation operation. Clearly, the second-order feature interactions adopted in Equation 3 could offer more expressive power to encode the hidden relevance between a viewpoint and an aspect [12, 17]. We also opt for including the latent representations of $\mathbf{v}_{u,x}$ and $\mathbf{a}_{i,y}$ to represent $g_{x,y}$. In our empirical evaluation, however, no further performance gain is observed.

3.2 Sentiment Capsule

Given there are M viewpoints/aspects for each user/item, we can form M^2 logic units in total with random pairing for each user-item pair. However, not all possible logic units are plausible or make sense in real-world. Here, we define *informative logic units* as the ones that are semantically plausible. That is, an informative logic unit should hold the explicit or implicit implication relation (or association) with respect to its constituent viewpoint and aspect in semantic sense. So one of our objective is to identify the informative

logic units. Furthermore, we target at inferring whether a user likes and dislikes an item according to what informative logic units, and to what extent. Here, we propose a capsule architecture based on [22] to implement all these objectives in one go.

In detail, two sentiment capsules, namely positive capsule and negative capsule, are adopted to jointly choose some logic units as the informative ones and resolve their sentiments. Specifically, in each sentiment capsule, the latent sentiment features are derived for each logic unit as $\mathbf{t}_{s,x,y} = \mathbf{W}_{s,x,y}\mathbf{g}_{x,y}$, where $s \in \mathcal{S}$ and $\mathcal{S} = \{\text{pos, neg}\}$ refers to the two sentiments, $\mathbf{W}_{s,x,y} \in R^{k \times 2k}$ is the weight matrix, and $\mathbf{t}_{s,x,y}$ is the latent feature vector for logic unit $g_{x,y}$. Then, the capsule with sentiment s takes all feature vectors $\{\mathbf{t}_{s,x,y}|x,y \in 1...M\}$ as input and derive its output vector via an iterative dynamic routing process. Actually, we can consider the output of each sentiment capsule as a weighted sum over these feature vectors:

$$\mathbf{s}_{s,u,i} = \sum_{x,y} c_{s,x,y} \mathbf{t}_{s,x,y} \tag{4}$$

where $c_{s,x,y}$ is the coupling coefficient indicating the importance of logic unit $g_{x,y}$ with respect to determining sentiment s (*i.e.,* like or dislike) and to what extent (ref. Equation 7). That is, a sentiment capsule is expected to capture a user's overall opinion in that sentiment. Besides, a capsule also encodes the probability of the existence (or activation) of the concept represented by the capsule in terms of the length of its output vector [22]. Specifically, a nonlinear squashing function transforms $\mathbf{s}_{s,u,i}$ into $\mathbf{o}_{s,u,i}$ whose length falls in the range of $(0, 1)$.

$$\mathbf{o}_{s,u,i} = \frac{\|\mathbf{s}_{s,u,i}\|^2}{1 + \|\mathbf{s}_{s,u,i}\|^2} \frac{\mathbf{s}_{s,u,i}}{\|\mathbf{s}_{s,u,i}\|} \tag{5}$$

where $\|\cdot\|$ denotes the length of a vector. Note that, in Equation 5, the orientation of $\mathbf{s}_{s,u,i}$ is retained in $\mathbf{o}_{s,u,i}$. We expect that $\mathbf{o}_{s,u,i}$ encodes the answer about to what extent user u likes or dislikes item i. Also, we utilize the length of $\mathbf{o}_{s,u,i}$ to answer whether user u holds sentiment s against item i in a probability perspective (*i.e.,* answer about whether user u likes or dislikes item i).
Limitation of Routing by Agreement. The value of $c_{s,x,y}$ in Equation 4 is calculated via an iterative Routing by Agreement

algorithm. As used in original capsule architecture [22], the core step is to iteratively calculate an agreement score $b_{s,x,y}$ indicating the relatedness between $\mathbf{t}_{s,x,y}$ and other feature vectors inside the same sentiment capsule. Then, a softmax operation is utilized to derive a logic unit's coupling coefficients with respect to the two capsules as follows:

$$c_{s,x,y} = \exp(b_{s,x,y}) / \sum_{s \in \mathcal{S}} \exp(b_{s,x,y}) \qquad (6)$$

In Equation 6, we can see that a logic unit will contribute itself more to the ouput of a sentiment capsule when it has relative more agreement with this capsule than the other one. Recall that we need to identify the informative logic units that represent the causes underlying a rating behavior. Based on Equation 6, a non-informative logic unit would still pose a large blur to hurt the sentiment abstraction ability of the capsules. For example, a non-informative logic unit would produce negative agreements (-0.05 vs. -0.9) with the two capsules at the begining of the dynamic routing process. However, with the softmax operation in Equation 6, the coupling coefficients are 0.7 vs. 0.3 respectively for this logic unit. The larger weight inevitably makes a lot of noise towards the first sentiment capsule, which causes a large adverse influence in later iterations (ref. Algorithm 1). In this sense, CARP fails to uncover the reasoning process as well as to guarantee semantic explanation. For this reason, we introduce Routing by Bi-Agreement.

Routing by Bi-Agreement. Let $\neg s$ denote the opposite sentiment compared to sentiment s, an appropriate dynamic routing process for our task should hold the following three properties: **(1)** Given $b_{s,x,y}$ is relative larger inside the capsule and $b_{s,x,y} > b_{\neg s,x,y}$, the value of $c_{s,x,y}$ should be larger as well; **(2)** Given $b_{s,x,y}$ is relative larger inside the capsule but $b_{s,x,y} < b_{\neg s,x,y}$, the value of $c_{s,x,y}$ should be small; **(3)** Given $b_{s,x,y}$ is relative smaller inside the capsule, the value of $c_{s,x,y}$ should be smaller also. Note that, given a user-item pair, the informative logic unit extracted from the related reviews would have different importance to explain a rating behavior. Also, each informative logic unit should be exclusively associated with a single sentiment. Here, we let CARP automatically identify the important informative logic units for a given user-item pair towards each sentiment, which is ruled by the first and second property in a review-driven manner. On the other side, non-informative logic units are expected to have very low agreement with both sentiment capsules, just as the example mentioned above. The third property could help us suppress their impact.

We propose Routing by Bi-Agreement (RBiA), a new iterative dynamic routing process that holds the above three properties. RBiA is devised to derive coupling coefficient $c_{s,x,y}$ by considering the relative magnitudes of $b_{s,x,y}$ in both inter-capsule and intra-capsule views together. The detail of this dynamic process is described in Algorithm 1. At first, RBiA assigns the same initial agreement for each logic unit over the two sentiment capsules (Lines 1-2). Then, we calculate a candidate coupling coefficient $\check{c}_{s,x,y}$ based on the inter-capsule comparison (Line 5). On the other hand, another candidate coupling coefficient $\hat{c}_{s,x,y}$ is calculated similarly, but by comparing $b_{s,x,y}$ in an intra-capsule view (Line 6). We calculate coupling cofficient $c_{s,x,y}$ by performing $L1$ normalization over the geometric mean of $\check{c}_{s,x,y}$ and $\hat{c}_{s,x,y}$ inside each sentiment capsule (Line 7). The geometric mean is an appealing choice for our task

---

Algorithm 1: *Routing by Bi-Agreement* Algorithm

input : Iteration number τ, feature vectors
$\qquad \mathcal{T} = \{\mathbf{t}_{s,x,y} | s \in \mathcal{S}, x \in 1...M, y \in 1...M\}$,
$\qquad \mathcal{S} = \{\text{pos}, \text{neg}\}$
output : Coupling coefficients $\{c_{s,x,y}\}$

1 **foreach** $\mathbf{t}_{s,x,y} \in \mathcal{T}$ **do**
2 $\quad \lfloor \ b_{s,x,y} = 0$;   /* Initialization */
3 **foreach** *iteration* **do**
4 $\quad$ **foreach** $\mathbf{t}_{s,x,y} \in \mathcal{T}$ **do**
$\qquad \quad$ /* A softmax over two sentiment capsules */;
5 $\qquad \quad \check{c}_{s,x,y} = \exp(b_{s,x,y}) / \sum_{s \in \mathcal{S}} \exp(b_{s,x,y})$;
$\qquad \quad$ /* A softmax inside a sentiment capsule */;
6 $\qquad \quad \hat{c}_{s,x,y} = \exp(b_{s,x,y}) / \sum_{j,k} \exp(b_{s,j,k})$;
$\qquad \quad$ /* $L1$ normalization   */;
7 $\qquad \quad c_{s,x,y} = \sqrt{\check{c}_{s,x,y} \hat{c}_{s,x,y}} / \sum_{j,k} \sqrt{\check{c}_{s,j,k} \hat{c}_{s,j,k}}$;
8 $\quad$ **foreach** $s \in \mathcal{S}$ **do**
9 $\qquad \quad \mathbf{s}_{s,u,i} = \sum_{x,y} c_{s,x,y} \mathbf{t}_{s,x,y}$;
10 $\qquad \quad \mathbf{o}_{s,u,i} = \text{squash}(\mathbf{s}_{s,u,i})$;   /* See Equation 5 */
11 $\quad$ **foreach** $\mathbf{t}_{s,x,y} \in \mathcal{T}$ **do**
$\qquad \quad$ /* Update the agreement   */;
12 $\qquad \quad b_{s,x,y} = b_{s,x,y} + \mathbf{t}_{s,x,y}^{\top} \mathbf{o}_{s,u,i}$;
13 **return** $\{c_{s,x,y}\}$

---

since a percentage change in either $\check{c}_{s,x,y}$ or $\hat{c}_{s,x,y}$ will result in the same effect on the geometric mean. By considering the two different views on agreement $b_{s,x,y}$, it is clear to see that RBiA can satisfy the three properties mentioned above. Note that each coupling coefficient $c_{s,x,y}$ is updated based on the measured agreement between feature vector $\mathbf{t}_{s,x,y}$ and the output $\mathbf{o}_{s,u,i}$ at the current iteration (Line 12). The output vector $\mathbf{o}_{s,u,i}$ calculated at the last iteration is then used for rating prediction, which will be described in the following.

3.3 Rating Prediction and Optimization

Given the output $\mathbf{o}_{s,u,i}$, we calculate rating $r_{s,u,i}$ user u would give for item i with sentiment s as follows:

$$r_{s,u,i} = \mathbf{w}_s^{\top} \mathbf{h}_{s,u,i} + b_{s,3} \qquad (7)$$

$$\eta_s = \sigma(\mathbf{H}_{s,1} \mathbf{o}_{s,u,i} + \mathbf{b}_{s,1}) \qquad (8)$$

$$\mathbf{h}_{s,u,i} = \eta_s \odot \mathbf{o}_{s,u,i} + (1 - \eta_s) \odot tanh(\mathbf{H}_{s,2} \mathbf{o}_{s,u,i} + \mathbf{b}_{s,2}) \qquad (9)$$

where $\mathbf{w}_s$ is the regression weight vector, $b_{s,3}$ is a bias term, $\mathbf{H}_{s,1}$, $\mathbf{H}_{s,2} \in R^{k \times k}$ are transform matrices, $\mathbf{b}_{s,1}, \mathbf{b}_{s,2} \in R^k$ are the bias vectors, and $\mathbf{1}$ is the vector with all elements being 1. By further considering the user and item biases, the overall rating score is then calculated as follows:

$$\hat{r}_{u,i} = f_C(r_{pos,u,i} \cdot \|\mathbf{o}_{pos,u,i}\| - r_{neg,u,i} \cdot \|\mathbf{o}_{neg,u,i}\|) + b_u + b_i \qquad (10)$$

where $\hat{r}_{u,i}$ is the predicted rating score, b_u and b_i are the corresponding bias for user u and item i respectively, and function $f_C(x) = 1 + \frac{C-1}{1+\exp(-x)}$ is a variant of the sigmoid function, producing a value within the target rating range of $[1, C]$. Here we

explicitly consider the presence of sentiment s for the given user-item pair in terms of vector length. Also, the utilization of one-layer highway network (ref. Equation 8 and 9) offers more flexibility to model complicated scenarios. For example, a user would dislike an item to some extent according to some cause . However, a high rating score would be also given by this user due to some more important causes (ref. Figure 1).

Multi-task Learning. For model optimization, we use Mean Square Error (MSE) as the objective to guide the parameter learning:

$$L_{sqr} = \frac{1}{|O|} \sum_{(u,i) \in O} (r_{u,i} - \hat{r}_{u,i})^2 \qquad (11)$$

where O denotes the set of observed user-item rating pairs, $r_{u,i}$ is the observed rating score assigned by user u to item i. Note that we explicitly include a particular sentiment analysis task in CARP. However, by matching the observed rating scores alone, the training of CARP would easily strap into a local optimum, which may not correctly reflect the fine-grained sentiments. Therefore, we introduce a sub-task to enhance the capacity of sentiment analysis in CARP. Specifically, a lower rating score suggests that the user certainly dislikes the item. Similarly, a higher score is rated by a user because of her most happiness on the item. Hence, we assign a label $s_{u,i}$ for each observed user-item rating pair. When rating $r_{u,i} > \pi$, $s_{u,i} = pos$; otherwise, $s_{u,i} = neg$. That is, we can split O into two disjoint subsets O_{pos} and O_{neg}. And π is a predefined threshold. We then utilize a sentiment analysis task as another objective function:

$$L_{stm} = \frac{1}{|O|} \Big(\sum_{(u,i) \in O} \max(0, \epsilon - \|\mathbf{o}_{s_{u,i},u,i}\|) \Big) \qquad (12)$$

where ϵ is a threshold indicating the probability that a user-item pair should hold for the corresponding sentiment. While Equation 12 guides the proposed CARP to learn the capacity of performing sentiment analysis more precisely, however, we need to consider the data imbalance issue. That is, the number of the user-item pairs with a lower rating score is significantly less than the couterpart (i.e., $|O_{pos}| \gg |O_{neg}|$). This imbalance problem hurts the model from identifying the negative sentiment. To recruit more supervision regarding the negative sentiment to alleviate this problem, we exploit a mutual exclusion principle as follows:

$$L_{stm} = \frac{1}{|O|} \Big(\sum_{(u,i) \in O} \max(0, \epsilon - \|\mathbf{o}_{s_{u,i},u,i}\|) + \max(0, \|\mathbf{o}_{\neg s_{u,i},u,i}\| - 1 + \epsilon) \Big) \qquad (13)$$

In Equation 13, each user-item pair should also meet the condition $\|\mathbf{o}_{\neg s_{u,i},u,i}\| \le 1 - \epsilon$. We know that this constraint is too strict to be true in reality. But more weak supervision about the negative sentiment is exploited for model optimization. In other words, the feedbacks regarding both the positive and negative sentiments are provided by each user-item pair. In our experiments, we observe that plausible results are gained by CARP with this principle during the prediction phase (ref. Section 4.3). In this work, we set $\epsilon = 0.8$. Considering both two objectives, the final objective for CARP is a linear fusion of L_{sqr} and L_{stm}:

$$L = \lambda \cdot L_{sqr} + (1 - \lambda) \cdot L_{stm} \qquad (14)$$

where λ is a tunable parameter controling the importance of each subtask. We use RMSprop [28] for parameter update in an end-to-end fashion.

4 EXPERIMENTS

In this section, comprehensive experiments are conducted on seven datasets from three different sources to evaluate the performance of CARP[1].

4.1 Experimental Setup

Datasets. In our experiment, we use seven datasets from three sources: Amazon-5cores[2] [10], Yelp, and the RateBeer website. For the Amazon-5cores dataset, five datasets from different domains are used (i.e., *Musical Instruments*, *Office Products*, *Digital Music*, *Tools Improvement*, and *Video Games*). The others are from the Yelp challenge website[3] (Round 11) and the RateBeer website[4] (called *Beer*), respectively. Note that, we also adopt 5-core settings over these two datasets: each user and item has at least 5 reviews. Besides, we only retain the records in Yelp spanning from 2016 to 2017 as the final dataset, denoted as *Yelp16-17*. The target rating ranges used in these datasets are $[1, 5]$[5], we therefore set $C = 5$ and $\pi = 3$.

On all datasets, we adopt the same preprocessing steps used in [15]. Then, we filter out the rating records which contain empty review afterwards. Detailed statistics of the seven preprocessed datasets are given in Table 1. We can see that ratio $|O_{pos}|/|O_{neg}|$ ("pos/neg ratio" in Table 1) is very large on most datasets. For each dataset, we randomly build the training set and testing set in the ratio 80 : 20. Moreover, 10% records in the training set are randomly selected as the validation set for hyper-parameter selection. Note that at least one interaction for each user/item is included in the training set. The target reviews in the validation and testing sets *are excluded* since they are unavailable in the practical scenario.

Baselines. Here, we compare the proposed CARP against the conventional baseline and recently proposed state-of-the-art rating prediction methods: (a) probabilistic matrix factorization that leverages only rating scores, **PMF** [23]; (b) latent topic and shallow embedding learning models with reviews, **RBLT** [26] and **CMLE** [37]; (c) deep learning based solutions with reviews, **DeepCoNN** [39], **D-Attn** [24], **TransNet** [3], **TARMF** [19], **MPCN** [27] and **ANR** [7]. Among these methods, D-Attn, TARMF, MPCN and ANR all identify important words for rating prediction. Note that there are many other state-of-the-art models, such as HFT [20], EFM [38], JMARS [8], CDL [30], ConvMF [15] and ALFM [6]. These works have been outperfomed by one or several baselines compared here. Hence, we omit further comparison for space saving.

Hyper-parameter Settings. We apply grid search to tune the hyper-parameters for all the methods based on the setting strategies reported in their papers. The final performances of all methods are reported over 5 runs. The latent dimension size is optimized from $\{25, 50, 100, 150, 200, 300\}$. The word embeddings are learned from scratch. The dimension size of word embedding is set to 300 (i.e., $d = 300$). The batch size for Musical Instruments, Beer, Office

---

[1]Our implementation is available at https://github.com/WHUIR/CARP.

[2]http://jmcauley.ucsd.edu/data/amazon/

[3]https://www.yelp.com/dataset/challenge

[4]https://www.ratebeer.com/

[5]In Beer, we convert the rating range of the Overall Rating which is $[4, 20]$ into $[1, 5]$.

Table 1: Statistics of the seven datasets

Datasets	# users	# items	# ratings	# words per review	# words per user	# words per item	pos/neg ratio	density
Musical Instruments	1,429	900	10,261	32.45	141.32	200.12	7.28	0.798%
Office Products	4,905	2,420	53,228	48.15	197.93	229.52	5.73	0.448%
Digital Music	5,540	3,568	64,666	69.57	216.21	266.51	4.14	0.327%
Tools Improvement	16,638	10,217	134,345	38.75	162.53	212.48	5.42	0.079%
Video Games	24,303	10,672	231,577	72.13	188.79	260.60	3.08	0.089%
Beer	7,725	21,976	66,625	17.31	34.06	103.20	1.47	0.039%
Yelp16-17	167,106	100,229	1,217,208	38.86	133.60	155.18	3.29	0.007%

Table 2: Overall performance comparison on seven datasets in terms of *MSE*. The best and second best results are highlighted in boldface and underlined respectively. ▲% denotes the relative improvement of CARP over the review-based alternatives. † indicates that the difference to the best result is statistically significant at 0.05 level.

Method	Musical Instruments	Office Products	Digital Music	Tools Improvement	Video Games	Beer	Yelp16-17
PMF	$1.398^\dagger$	$1.092^\dagger$	$1.206^\dagger$	$1.566^\dagger$	$1.672^\dagger$	$1.641^\dagger$	$2.574^\dagger$
RBLT	$0.815^\dagger$	$0.759^\dagger$	$0.870^\dagger$	$0.983^\dagger$	$1.143^\dagger$	$\underline{0.576}^\dagger$	$1.569^\dagger$
CMLE	$0.817^\dagger$	$0.759^\dagger$	$0.885^\dagger$	$1.020^\dagger$	$1.253^\dagger$	$0.605^\dagger$	$1.593^\dagger$
DeepCoNN	$0.814^\dagger$	$0.860^\dagger$	$1.056^\dagger$	$1.061^\dagger$	$1.238^\dagger$	$0.618^\dagger$	$1.593^\dagger$
D-Attn	$0.982^\dagger$	$0.825^\dagger$	$0.911^\dagger$	$1.043†$	$\underline{1.145}^\dagger$	$0.614^\dagger$	$1.573^\dagger$
TransNet	$0.798^\dagger$	$0.759^\dagger$	$0.913^\dagger$	$1.003^\dagger$	$1.190^\dagger$	$0.587^\dagger$	$\underline{1.523}^\dagger$
TARMF	$0.943^\dagger$	$0.789^\dagger$	$\underline{0.853}^\dagger$	$1.169^\dagger$	$1.195^\dagger$	$0.912^\dagger$	$1.914^\dagger$
MPCN	$0.824^\dagger$	$0.769^\dagger$	$0.903^\dagger$	$1.017^\dagger$	$1.201^\dagger$	$0.616^\dagger$	$1.617^\dagger$
ANR	$\underline{0.795}^\dagger$	$\underline{0.742}^\dagger$	$0.867^\dagger$	$\underline{0.975}^\dagger$	$1.182^\dagger$	$0.590^\dagger$	$1.553^\dagger$
CARP	**0.773**	**0.719**	**0.820**	**0.960**	**1.084**	**0.556**	**1.508**
▲%	$2.8 - 21.3$	$3.1 - 16.4$	$3.9 - 22.3$	$1.5 - 17.9$	$5.3 - 13.5$	$3.5 - 39.0$	$0.98 - 21.2$
CARP-RA	0.789	0.727	0.836	0.969	1.100	0.567	1.536

Table 3: Impact of different viewpoint/aspect numbers in CARP. The best results are highlighted in boldface.

M	Musical Instruments	Office Products	Digital Music	Tools Improvement	Video Games	Beer	Yelp16-17
3	0.776	0.728	0.824	0.961	1.090	0.560	1.513
5	0.776	0.716	**0.818**	0.962	1.087	**0.552**	**1.503**
7	**0.771**	**0.715**	**0.818**	0.958	1.083	0.564	1.517
9	0.778	0.726	0.819	**0.950**	**1.074**	0.562	1.509

Products and Digital Music is set to 100. For other bigger datasets, the batch size is set to 200. The number of convolution filters is set to 50 for convolution based methods (including CARP, $n = 50$).

For CARP, window size is $c = 3$, the iteration number τ is set to 3 for Routing by Bi-Agreement, the number of viewpoints or aspects for each user/item is set to be 5 (*i.e.*, $M = 5$), and λ is 0.5. We set the keep probability of dropout to be 0.9 and learning rate to 0.001 for model training. The dimension size k is set to be 25^6.
Evaluation Metric. Here, we use *MSE* (ref. Equation 11) as performance metric, which is widely adopted in many related works for performance evaluation [19, 27, 32]. The statistical significance test is conducted by performing the student *t-test*.

---

[6] We observe that k is optimal in [25, 100] on all datasets.

4.2 Performance Evaluation

A summary of results of all methods over the seven datasets are reported in Table 2. Several observations can be made. First, it is not surprising that the interaction-based method (PMF) consistently yields the worst on all seven datasets. This observation is consistent with what have been made in many review-based works [15, 24, 39].

Second, among review-based baselines, there is no dominating winner acorss seven datasets. DeepCoNN consistently performs worse than the other solutions across different datasets. This is reasonable since neither word-level nor aspect-level attention mechanism is used for the feature extraction. In contrast, a much better performance is obtained by D-Attn against DeepCoNN in most datasets, due to the dual word-level attention mechanisms utilized in the former. Also, TransNet performs significantly better than DeepCoNN in all the datasets. This observation is consistent with

(a) Ratio values on different ranks   (b) Performance on differnt λ values
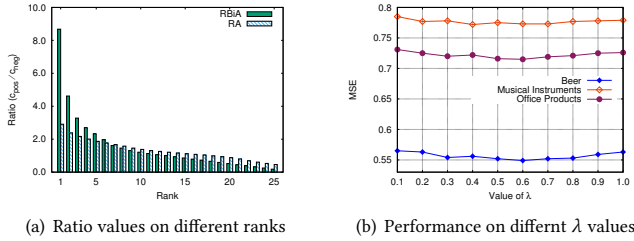
Figure 3: Ratios at different ranks (a), and performance with λ values (b).

Table 4: Impact of different numbers of iterations in RBiA. The best results are highlighted in boldface.

τ	Musical Instruments	Office Products	Digital Music	Tools Improvement	Beer
1	0.786	0.733	0.852	0.976	0.573
2	0.778	0.720	0.833	0.963	0.557
3	0.776	**0.716**	**0.818**	0.962	**0.552**
4	**0.772**	0.727	0.827	**0.954**	0.555

prior works [3, 27]. By repeating each review according to its rating score (*i.e.,* rating boosting), RBLT can precisely extract the semantic topics from the textual information. Relative good performance (*i.e.,* comparable to the best baseline) is achieved across different datasets, especially on the sparser ones. ANR performs the best in three datasets while TransNet, D-Attn, TARMF and RBLT outperform the others in each of the rest four datasets. Note that ANR models semantic information from reviews in the aspect-level. This suggests that the fine-grained representation learning is a promising avenue towards better understanding of a rating behavior.

Third, as Table 2 shows, CARP consistently achieves the best performance across the seven datasets. It is worthwhile to highlight that a significant improvement is gained by CARP on Beer which is the second sparsest dataset with least review information (ref. Table 1). Compared with three recently proposed state-of-the-art models (*i.e.,* TARMF, MPCN and ANR), CARP obtains up to 39.0%, 9.7% and 8.0% relative improvement respectively. Overall, the experimental results demonstrate that CARP is effective in modeling a rating behavior from reviews for rating prediction.

4.3 Analysis of CARP

We further investigate the impact of the parameter settings (*i.e.,* τ, M, λ) to the performance of CARP on the validation set. When studying a parameter, we fix the other parameters to the values described in Section 4.1. Note that not all performance patterns in all datasets are presented below. Since the similar pattern is observed and performance scales are quite different, we omit some of them for space saving.

Impact of Dynamic Routing. We investigate the impact of τ in the dynamic routing of CARP. Table 4 reports the performance pattern of CARP with varying iteration numbers in terms of *MSE*. It is clear that more than two iterations leads to better prediction accuracy. The optimal performance is obtained by performing either 3 or 4 iterations. Based on the result, we perform three iterations in CARP (*i.e.,* $\tau = 3$). We also conduct the experiments with the

standard Routing by Agreement (RA) as proposed in [22]. Last row of Table 2 reports the performance over seven datasets with this setting (*i.e.,* CARP-RA). It is found that the performance deteriorates to different extent. For each logic unit $g_{x,y}$ ranked at i-th position in the positive capsule, we further check the corresponding ratio $c_{pos,x,y}/c_{neg,x,y}$. Figure 3(a) reports the averaged ratio for each position in the positive capsule calculated by RBiA and RA respectively over the testing set of Musical Instruments. We can see that the coupling coefficients of logic units calculated by RBiA are more sharp. This suggests that RBiA is more appropriate to identify the informative logic units that are highly relevant to a user's rating. Similar patterns are also observed in the other datasets. Overall, the results suggest the superiority of the proposed RBiA by suppressing the adverse impact from the non-informative logic units.

Impact of M Value. The M value specifies the number of viewpoints/aspects extracted for each user/item respectively. Here, we report the performance patterns of CARP by tuning M amongst $\{3, 5, 7, 9\}$. As shown in Table 3, the optimal M value is not consistent across the datasets. It seems that $M = 5/7$ is desired in more datasets. Given the performance variation is small, we choose to use $M = 5$ in our experiments.

Impact of Multi-task Learning. Recall in Equation 14, parameter λ controls the tradeoff in the multi-task learning setting of CARP. Figure 3(b) plots the performance patterns of three datasets by tuning λ in the range of $[0.1, 1]$ with a step of 0.1. When $\lambda = 0$, there is not supervision towards how to predict the final rating scores. We observe that the optimal performance is consistently achieved when λ is around 0.5. We also observe that the performance becomes increasingly worse when $\lambda \to 1$. This validates the positive benefit of the proposed multi-task learning process in CARP. Based on the results, we fix λ to be 0.5 in our experiments, though this may not be an optimal setting for some datasets. Note that we also introduce a mutual exclusion principle in the multi-task learning process (ref. Equation 12 and 13). Without this mutual exclusion, CARP produces a worse performance which is comparable with the setting of $\lambda = 1$.

4.4 Explainability Analysis

We further check whether CARP can discover proper causes and their effects to interpret a rating behavior.

Table 5 displays these sentences and the auxiliary information provided by CARP. Recall that CARP has encoded contextual information to represent each word. To better visualize a viewpoint (an aspect), we retrieve the top-K phrases whose weight is the sum of the weights (*i.e., $attn_{u,x,j}$* in Equation 2) of the constituent words in the convolutional window. Finally, we pick the sentences containing these informative phrases to represent the corresponding viewpoint/aspect. Here, we choose $K = 30$. We randomly sample three user-item pairs from two datasets. The first two pairs with the same user but different items are from Office Products. The last one is picked from Musical Instruments. For a pair with a higher rating, we list top-2 and top-1 logic units extracted by positive and negative sentiment capsules respectively, and vice versa. Note that an extracted viewpoint (or aspect) could cover several sub-viewpoints (or sub-aspects), given our M is a global parameter. Also, it is observed that some phrases would appear in two viewpoints or aspects. This is reasonable since a viewpoint or aspect is a high-level concept

Table 5: Example study of three user-item pairs from Office Products and Musical Instruments.

user₁ - item₁ (Smead MO File Box): $r_{1,1} = 2.0$, $\hat{r}_{1,1} = 3.03$, $\|o_{pos,1,1}\| = 0.589$, $\|o_{neg,1,1}\| = 0.460$, $r_{pos,1,1} = 0.761$, $r_{neg,1,1} = 0.735$			$c_{pos,x,y}$	$c_{neg,x,y}$
$g_{2,2}$	viewpoint:	The base is solid metal with high quality casters.	0.034	**0.079**
	aspect:	One last thing, I think is holds light stuffs, don't over loading it.		
$g_{1,5}$	viewpoint:	These labels save me money by letting me reuse old folders.	0.026	**0.064**
	aspect:	My concerns is the price. I think is seriously overprice for a cardboard box. ···		
$g_{5,3}$	viewpoint:	These are great organizational tools, whether to replace lost tabs or to relabel a whole drawer in your own unique manner.	**0.074**	0.029
	aspect:	I could easily use a bunch of these to organize papers and magazines at work and at home		
target review		··· The Smead MO File Box seemed like it could be a good way to keep those papers right at hand. ··· It's a little disapointing for the price that it doesn't come with a slip-cover. ··· The overall would not recommend it for general use unless they reassess the price.		
user₁ - item₂ (Post-It Big Pad): $r_{1,2} = 4.0$, $\hat{r}_{1,2} = 4.45$, $\|o_{pos,1,2}\| = 0.726$, $\|o_{neg,1,2}\| = 0.293$, $r_{pos,1,2} = 0.737$, $r_{neg,1,2} = 0.651$			$c_{pos,x,y}$	$c_{neg,x,y}$
$g_{3,4}$	viewpoint:	It is also visually attractive with a silver back support, and would look great in any office.	**0.078**	0.027
	aspect:	Positives: Vivid yellow and large footprint and ability to stick to surfaces means your message will be received.		
$g_{1,1}$	viewpoint:	Each tab has adhesive that holds well to the page but can also be removed if needed.	**0.071**	0.017
	aspect:	The adhesive is wide enough to securely hold the paper on many different surfaces		
$g_{2,1}$	viewpoint:	The base is solid metal with high quality casters.	0.018	**0.083**
	aspect:	I want to cover some of my notes I had pre-written on a white board.		
target review		··· The stick-em doesn't appear any stickier than a standard post-it, so I suspect these extra large strips are needed to hold it on the wall. ··· These large pads will definitely serve a purpose in these sessions, allowing larger print that is easier to read.		
user₂ - item₃ (Guitar Patch Cable): $r_{2,3} = 5.0$, $\hat{r}_{2,3} = 4.56$, $\|o_{pos,2,3}\| = 0.759$, $\|o_{neg,2,3}\| = 0.295$, $r_{pos,2,3} = 0.941$, $r_{neg,2,3} = 0.753$			$c_{pos,x,y}$	$c_{neg,x,y}$
$g_{1,4}$	viewpoint:	I found that these were OK but the heads were bigger than I could stand to use on my cramped for space pedal board.	**0.124**	0.0198
	aspect:	I use these patch cables on my pedal board and they save valuable space		
$g_{4,3}$	viewpoint:	The delay pedal was so cheap that I could not pass it up. And let me tell you, it is easy and fun to use	**0.090**	0.035
	aspect:	Good quality in spite of their reasonable price.		
$g_{3,1}$	viewpoint:	Love the heavy cord and gold connectors.	0.019	**0.065**
	aspect:	Plus, they look nice, unlike the various colors that come with the bulkier, cheaper 1'cords		
target review		compact heads allow more pedals to be loaded onto your overpriced pedal board. the quality is good. ···		

composited by different phrases, some of which have just general semantics. We choose one or two of the most informative sentences of each logic unit. The informative phrases are highlighted by orange color. The stop words inside a context is also highlighted for better understand. The blue color indicates that the phrase is associated with more than one viewpoint/aspect. We also transform raw $r_{s,u,i}$ value by Max-Normalization to ease interpretation, since the final rating is estimated through a nonlinear transfer (ref. Equation 10). As a reference, we display the parts matched well by these logic units in target user-item review with red and green underlines for positive and negative sentiment capsules respectively.

Example 1: the item in the first pair is a Smead MO File Box with a low rating (*i.e.,* 2.0) from the user. The first logic unit suggests that the user prefers metal texture and high quality. However, the associated aspect shows that this item can not hold heavy things. The second logic unit directly indicates that the user is a saver and the item is obviously overprice. There is also some thing the user likes about this item. The third logic unit shows that the user loves the flexiblity and versatility, which is indeed expressed in item document. However, this is a weak positive effect. Note that the causes and effects expressed by logic units $g_{1,5}$ and $g_{5,3}$ are really mentioned in the target review.

Example 2: compared with the first pair, the second one illustrates the causes and effects towards a Post-it Big Pads by the same user with a high rating (*i.e.,* 4.0). The first and second logic units suggest that some information needs (*i.e.,* attractive appearance and convenient to post something) are matched well by the item. The target review written by the user also covers these two logic units very well. The third logic unit is not straightforward to interpret. Some implication, however, indeed exists underlying this logic unit. As displayed by the second logic unit in the first example, this user mentions metal texture and high quality (the same viewpoint in two logic units), indicating she also likes solid and invulnerable things. Meanwhile, it is true that a Post-it note does not have the

property. The short length of the output of the negative capsule (*i.e.,* 0.293) indicates that this cause has a weak negative effect, which is also confirmed by the overall high rating from the user.

Example 3: this example is from Musical Instruments. The first logic unit shows that the user-item pair reaches an agreement on space saver. Similarly, the second logic unit demonstrates that the user holds a cost-effective viewpoint which is well matched by the high cost performance of the item. This logic unit is also well matched in the target review. The third logic unit involves a bit reasoning. It is obvious that the user prefers something with cool appearance. In contrast, this item looks uninteresting based on the review. The high rating indicates that this logic unit is not important. This effect is also captured by CARP with a relative lower $r_{neg,u,i}$ (*i.e.,* 0.753) and a small vector length (*i.e.,* 0.295).

Overall, our results demonstrate that CARP is superior in rating prediction and facilitates explanation with causes and effects at a finer level of granularity.

5 CONCLUSION

The proposed capsule network based model for recommendation and explanation can be seen as a single step towards reasoning a particular kind of human activity. Our extensive experiments demonstrate that better recommendation performance and understanding are obtained by modeling the causes and effects in fine-grained level. Actually, reasoning a rating behavior in terms of user viewpoints, item aspects and sentiments could nourish further benefits in E-Commerce, including cross-domain recommendation, user profiling, and user-oriented review summarization. We will invesigate these possibilities in the future.

ACKNOWLEDGMENTS

REFERENCES

[1] Yang Bao, Hui Fang, and Jie Zhang. 2014. TopicMF: Simultaneously Exploiting Ratings and Reviews for Recommendation. In *AAAI*. 2–8.
[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.
[3] Rose Catherine and William W. Cohen. 2017. TransNets: Learning to Transform for Recommendation. In *RecSys*. 288–296.
[4] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-level Explanations. In *WWW*. 1583–1592.
[5] Zhiyong Cheng, Ying Ding, Xiangnan He, Lei Zhu, Xuemeng Song, and Mohan S. Kankanhalli. 2018. Aˆ3NCF: An Adaptive Aspect Attention Model for Rating Prediction. In *IJCAI*. 3748–3754.
[6] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan S. Kankanhalli. 2018. Aspect-Aware Latent Factor Model: Rating Prediction with Ratings and Reviews. In *WWW*. 639–648.
[7] Jin Yao Chin, Kaiqi Zhao, Shafiq Joty, and Gao Cong. 2018. ANR: Aspect-based Neural Recommender. In *CIKM*. 147–156.
[8] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *KDD*. 193–202.
[9] Jeffrey L. Elman. 1990. Finding Structure in Time. *Cognitive Science* 14, 2 (1990), 179–211.
[10] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *WWW*. 507–517.
[11] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. TriRank: Review-aware Explainable Recommendation by Modeling Aspects. In *CIKM*. 1661–1670.
[12] Xiangnan He and Tat-Seng Chua. 2017. Neural Factorization Machines for Sparse Predictive Analytics. In *SIGIR*. 355–364.
[13] Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. 2011. Transforming Auto-Encoders. In *ICANN*. 44–51.
[14] Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *SIGIR*. 50–57.
[15] Dong Hyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional Matrix Factorization for Document Context-Aware Recommendation. In *RecSys*. 233–240.
[16] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*. 1746–1751.
[17] Chenliang Li, Wei Zhou, Feng Ji, Yu Duan, and Haiqing Chen. 2018. A Deep Relevance Model for Zero-Shot Document Filtering. In *ACL*. 2300–2310.
[18] Guang Ling, Michael R. Lyu, and Irwin King. 2014. Ratings meet reviews, a combined approach to recommend. In *RecSys*. 105–112.
[19] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Coevolutionary Recommendation Model: Mutual Learning between Ratings and Reviews. In *WWW*. 773–782.
[20] Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*. 165–172.
[21] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proc. of INTERSPEECH*. 1045–1048.
[22] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing Between Capsules. In *NIPS*. 3859–3869.
[23] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *NIPS*. 1257–1264.
[24] 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 *RecSys*. 297–305.
[25] Kaisong Song, Wei Gao, Shi Feng, Daling Wang, Kam-Fai Wong, and Chengqi Zhang. 2017. Recommendation vs Sentiment Analysis: A Text-Driven Latent Factor Model for Rating Prediction with Cold-Start Awareness. In *IJCAI*. 2744–2750.
[26] Yunzhi Tan, Min Zhang, Yiqun Liu, and Shaoping Ma. 2016. Rating-Boosted Latent Topics: Understanding Users and Items with Ratings and Reviews. In *IJCAI*. 2640–2646.
[27] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-Pointer Co-Attention Networks for Recommendation. In *KDD*. 2309–2318.
[28] Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* 4, 2 (2012), 26–31.
[29] Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *KDD*. 448–456.
[30] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative Deep Learning for Recommender Systems. In *KDD*. 1235–1244.
[31] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable Recommendation via Multi-Task Learning in Opinionated Text Data. In *SIGIR*. 165–174.
[32] Libing Wu, Cong Quan, Chenliang Li, and Donghong Ji. 2018. PARL: Let Strangers Speak Out What You Like. In *CIKM*. 677–686.
[33] Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu. 2018. Zero-shot User Intent Detection via Capsule Neural Networks. In *EMNLP*. 3090–3099.
[34] Liqiang Xiao, Honglun Zhang, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2018. MCapsNet: Capsule Network for Text with Multi-Task Learning. In *EMNLP*. 4565–4574.
[35] Min Yang, Wei Zhao, Jianbo Ye, Zeyang Lei, Zhou Zhao, and Soufei Zhang. 2018. Investigating Capsule Networks with Dynamic Routing for Text Classification. In *EMNLP*. 3110–3119.
[36] Ningyu Zhang, Shumin Deng, Zhanling Sun, Xi Chen, Wei Zhang, and Huajun Chen. 2018. Attention-Based Capsule Network with Dynamic Routing for Relation Extraction. In *EMNLP*. 986–992.
[37] Wei Zhang, Quan Yuan, Jiawei Han, and Jianyong Wang. 2016. Collaborative Multi-Level Embedding Learning from Reviews for Rating Prediction. In *IJCAI*. 2986–2992.
[38] 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 *SIGIR*. 83–92.
[39] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *WSDM*. 425–434.