Dynamic Review-based Recommenders

Kostadin Cvejoski*‡, Ramsés J. Sánchez*†, Christian Bauckhage‡ and César Ojeda§

*Competence Center Machine Learning Rhine-Ruhr, 53757 Sankt Augustin, Germany
†University of Bonn, 53113 Bonn, Germany
‡Fraunhofer Center for Machine Learning and Fraunhofer IAIS, 53757 Sankt Augustin, Germany
§Berlin Center for Machine Learning and TU Berlin, 10587 Berlin, Germany
{kostadin.cvejoski, christian.bauckhage}@iais.fraunhofer.de,
ojeda.marin@tu-berlin.de, sanchez@bit.uni-bonn.de

Abstract—**Just as user preferences change with time, item reviews also reflect those same preference changes. In a nutshell, if one is to sequentially incorporate review content knowledge into recommender systems, one is naturally led to dynamical models of text. In the present work we leverage the known power of reviews to enhance rating predictions in a way that (i) respects the causality of review generation and (ii) includes, in a bidirectional fashion, the ability of ratings to inform language review models and vice-versa, language representations that help predict ratings end-to-end. Moreover, our representations are time-interval aware and thus yield a continuous-time representation of the dynamics. We provide experiments on real-world datasets and show that our methodology is able to outperform several state-of-the-art models. Source code for all models can be found at [1].**

Index Terms—**recurrent recommender networks, dynamic language model, attention for recommendation**

I. INTRODUCTION

Following the deep learning agenda, the success of modern recommender systems heavily relies on their ability to leverage meaningful representations that allow for the accurate prediction of a purchase or a rating. Fundamentally one must dwell into *interest modeling*, as an effective recommendation is such that it uncovers, for a given user, a hidden interest in an unknown item. It is natural to study the change of user interest with time and, in the present work, we seek to incorporate these dynamical notions with those of text reviews.

Reviews are effectively a form of recommendation, and one that is directly provided by the user. The challenge however, stems from the unstructured and ambiguous nature of reviews (and natural language itself). A user might, for example, simultaneously highlight positive and negative aspects of the different items she reviews. Following current trends in natural language processing, we leverage review content through neural models of text and attention mechanisms, and guarantee the information content of those representations via reproduction quality. We encourage the dynamical aspect of text recommendations by learning representations which help predict both *when* is the next review arriving and *what* does it say. One is then led naturally to dynamical language models, since enforcing good text predictions ensures its dynamical representation quality.

II. RELATED WORK

There is a large body of research invested in recommender systems (RS), a big part of which has lately been devoted to capture the temporal dynamics of both users and items. One of the first temporal models for recommendation is the TimeSVD++ [2], which extends the SVD++ matrix factorization algorithm by introducing time-dependent latent factors. From the neural network perspective, many models for RS have been developed [3]–[5], and Recurrent Neural Networks (RNNs) have been used to capture time-ordered user activity. For example, session-based item recommendation use RNNs to infers user preferences from sessions of user behaviour [6]–[9]. Another example, closer to our work, is the Recurrent Recommender Networks (RRN) which uses two independent RNNs to model user and item dynamics separately [10].

Just as with user (and item) temporal representations, including review content representations has also been shown to significantly improve rating prediction and item recommendation [11]–[14]. However, some of these models break *causality*, in the sense that they either use the review of the item whose rate one is predicting, or use item reviews that have not been received by the time the item of interested was rated.

Finally, a model that combines RRN (a dynamical RS) with character-based autoregressive language models for reviews has recently been develop [15]. This work however, does not leverage the review content for rating prediction.

In contrast to all these works, we combine dynamical recommender systems with a dynamical language model that captures review content evolution, and use the review representations together with the user-item temporal representations in a causal fashion, to predict the rating of the next review.

III. DYNAMIC REVIEW-BASED RECOMMENDERS (DRR)

The interests and preferences of users vary as they age, or change their social status or lifestyle. Exogenous factors like trends or seasons also affect user preferences. For example, users tend to look for different clothe types in winter than those they look for in summer. Users also tend to change their music tastes as they age. Such preference changes are naturally encoded in the collections of reviews and ratings given by these users over time. Our goal is to learn representations capturing them. We therefore develop a model that explicitly
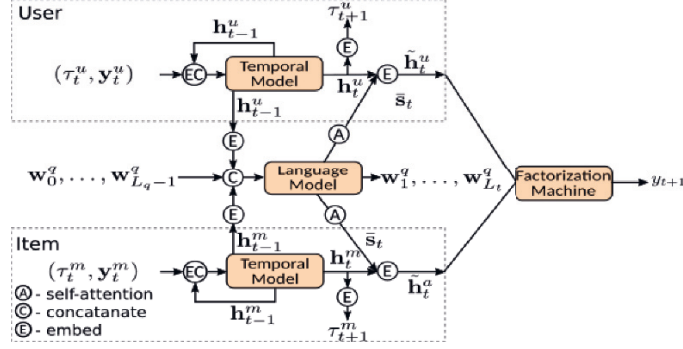
Fig. 1: Dynamic Review-based Recommender. The model consists of three interacting components: (i) a temporal model composed of two RNNs, one for users and the other for items, which we called *Dynamic Model of Review Sequences*; (ii) a neural language model which leverages the temporal representations of both user and items, and which we called *Dynamic Model of Review Content*; and (iii) a *Rating Model* which combines the user and item temporal representations with the review content representations to predict ratings. Note that when $q = t$ in the language model component, the Dynamic Review-based Recommender is causal. The model is non-causal when $q = t + 1$.

uses the text content and ratings of past reviews together with the history of when those reviews were written to better predict user interest in unknown items.

Consider a dataset $\mathcal{D}$ with a number of V items (as e.g. businesses or services, movies, products, etc.) and a number of U users. An element $e \in \mathcal{D}$ consists of a sequence of N_e reviews $\mathbf{r}_e = \{(\mathbf{x}_t^e, \tau_t^e, \delta_t^e, \mathbf{y}_t^e)\}_{t=1}^{N_e}$, where the t-th review is composed of its text $\mathbf{x}_t^e$, creation time τ_t^e, inter-review time $\delta_t^e \equiv \tau_t^e - \tau_{t-1}^e$ and rating vector $\mathbf{y}_t^e$.

Such review sequences $\mathbf{r}_e$ effectively define time series, and each of these can either be associate with a user u (in which case we set $e = u$), or an item v (in which case $e = v$).

Thus the rating vector for user u is such that $\mathbf{y}_t^u \in \mathbb{R}^V$, with $\mathbf{y}_{t,v}^u = p$ if user u rated item v with rating p. Conversely, the rating vector for item v is such that $\mathbf{y}_t^v \in \mathbb{R}^U$. Note that both of these vectors are large and sparse. To process them efficiently we perform dimensionality reduction via hashing, following [16].

Our main idea is to model the user and item review sequences separately, via two independent RNNs which output temporal representations encoding the nonlinear relations between timing and rating of past reviews. We then feed these temporal representations to neural models of text, thereby yielding instantaneous review content models, while simultaneously use them to predict when are new reviews going to arrive and what are their ratings. The model thus consists of tree interacting components: a temporal model composed of two RNNs, one for users and the other for items, which we called *Dynamic Model of Review Sequences*, a neural language model which leverages the temporal representations of both user and items, and which we called *Dynamic Model of Review Content*, and a *Rating Model* which combines the user and item temporal representations with the review content representations to predict ratings. In what follows we dwell into the details of these building blocks. Figure 1 summarizes the Dynamic Review-based Recommender (DRR) model.

A. Dynamic Model of Review Sequences

Given a sequence of reviews $\mathbf{r}_e$, we process each of its elements recursively via a RNN with hidden state $\mathbf{h}_t^e \in \mathbb{R}^H$. At each timestep t, we first compute the hidden representation

$$\mathbf{z}_t^e = \mathbf{W}_\tau^e \tau_t^e + \mathbf{W}_\delta^e \delta_t^e + \mathbf{W}_y^e \mathbf{y}_t^e + \mathbf{b}^e, \tag{1}$$

where $\mathbf{W}_\tau^e, \mathbf{W}_\delta^e, \mathbf{W}_y^e$ and $\mathbf{b}^e$ are learnable parameters and $\mathbf{z}_t^e \in \mathbb{R}^E$. We then update the RNN's hidden state thus

$$\mathbf{h}_t^e = f_\theta^{(e)}(\mathbf{z}_t^e, \mathbf{h}_{t-1}^e), \tag{2}$$

where $f_\theta^{(e)}$ is implemented by a LSTM network [17].

Note that the superindex e is used here to emphasize that we have two sets of functions namely, one for the user ($e = u$) and one for the item ($e = v$) reviews. The temporal representation $\mathbf{h}_t^e$ thus defined not only encodes the history of ratings, but also the time lag between past reviews, thereby yielding a continuous-time representation of the dynamics.

To enforce encoding quality, we first use $\mathbf{h}_t^e$ to predict the arrival time of new reviews via a simple Review Creation Model, which we shall now introduce. Later we will explicitly use $\mathbf{h}_t^u$ and $\mathbf{h}_t^v$ to predict ratings through a Rating Model.

1) Review Creation Model: The inter-review times δ_t^e can be modeled as following an exponential distribution whose rate parameter $\lambda_\theta^{(e)}(\mathbf{h}_t^e)$ is a function of the temporal representation $\mathbf{h}_t^e$ [18], [19]. In practice we approximate the function $\lambda_\theta^{(e)} : \mathbb{R}^H \to \mathbb{R}_{>0}$ with a multi-layer perceptron. The log-likelihood of the Review Creation Model is then

$$\begin{aligned}
\log p(\delta^e) &= \sum_{t=1}^{N_e} \log p_\theta(\delta_{t+1}^e | \mathbf{h}_t^e) \\
&= \sum_{t=1}^{N_e} \left(\log \lambda_\theta^{(e)}(\mathbf{h}_t^e) - \lambda_\theta^{(e)}(\mathbf{h}_t^e) \delta_{t+1}^e \right).
\end{aligned} \tag{3}$$

Note that predicting the arrival times of new reviews can be done by either sampling the exponential distribution, or using

the mean of the distribution directly. In our experiments we use the mean of the distribution.

B. Dynamic Model of Review Content

Consider the t-th review in the sequence $\mathbf{r}_e$, whose text content is given by $\mathbf{x}_t^e = (\mathbf{w}_0^{e,t}, \mathbf{w}_1^{e,t}, \ldots, \mathbf{w}_{L_t^e}^{e,t})$, where $\mathbf{w}_j^{e,t}$ and L_t^e label the j-th word and the number of words in that review, respectively. To capture how the review content changes within $\mathbf{r}_e$, we define the probability of observing the word sequence $\mathbf{x}_t^e$ at the t-th review as the conditional probability $p(\mathbf{x}_t^e | \mathbf{h}_{t-1})$. Here we define the global temporal representation $\mathbf{h}_t$ encoding the nonlinear relations between timing and ratings of past reviews as $\mathbf{h}_t \equiv \text{concat}([\mathbf{h}_t^u, \mathbf{h}_t^v])$, with $\mathbf{h}_t^{u,v}$ defined in Eq. 2.

Note that when processing the dataset $\mathcal{D}$, the modeling of review content does not need to differentiate between user and item. We therefore drop the superindex e in what follows.

Below we present two models for $p(\mathbf{x}_t | \mathbf{h}_{t-1})$, one based on a Bag-of-Words (BoW) representation, and another on an autoregressive language model. Both models will be trained by maximising $\log p(\mathbf{x}_t | \mathbf{h}_{t-1})$. These language models will ultimately allow us to define a vector representation $\bar{\mathbf{s}}_t$, summarizing the content of the t-th review, which we will later use as input to our Rating Model.

1) Bag-of-Words Neural Review Model: We assume the words in $\mathbf{x}_t$ are generated independently, conditioned on $\mathbf{h}_{t-1}$, that is $p(\mathbf{x}_t | \mathbf{h}_{t-1}) = \prod_j^{L_t} p_\theta(\mathbf{w}_j^t | \mathbf{h}_{t-1})$, where we follow [20] and write the probability over words as

$$p_\theta(\mathbf{w}_j^t | \mathbf{h}_{t-1}) = \frac{\exp\{-a(\mathbf{w}_j^t, \mathbf{h}_{t-1})\}}{\sum_{k=1}^V \exp\{-a(\mathbf{w}_k^t, \mathbf{h}_{t-1})\}},$$
$$a(\mathbf{w}_j^t, \mathbf{h}_{t-1}) = -\mathbf{h}_{t-1}^\top \mathbf{R}\,\mathbf{w}_j^t - \mathbf{b}\,\mathbf{w}_j^t, \tag{4}$$

with $\mathbf{R} \in \mathbb{R}^{2H \times V}$ and $\mathbf{b} \in \mathbb{R}^V$ trainable parameters, $\mathbf{h}_t = \text{concat}([\mathbf{h}_t^u, \mathbf{h}_t^v])$ and $\mathbf{w}_j^t$ the one-hot representation of the j-th word in $\mathbf{x}_t$.

We define the summary representation for $\mathbf{x}_t$ as the Bag-of-Words (BoW) representation $\bar{\mathbf{s}}_t \in \mathbb{R}^V$, where V is the vocabulary size [21].

2) Autoregressive Review Model: In contrast to the BoW model above, autoregressive language models approximate the probability over the word sequence $\mathbf{x}_t$ as [22]

$$p(\mathbf{x}_t | \mathbf{h}_{t-1}) = \prod_{j=1}^{L_t} p_\theta(\mathbf{w}_j^t | \mathbf{w}_{<j}^t, \mathbf{h}_{t-1}), \tag{5}$$

where $\mathbf{w}_{<j}^t$ labels all words previous to $\mathbf{w}_j^t$.

The conditional probability above depends on both $\mathbf{w}_{<j}^t$ and the global temporal representation $\mathbf{h}_t$. To model it we take an approach akin to that of the variational autoencoders of text [23]. That is, we first concatenate $\mathbf{h}_t$ with all word embeddings in $\mathbf{x}_t$, i.e. we define $\tilde{\mathbf{w}}_j = \text{concat}[\mathbf{w}_j, \mathbf{h}_{t-1}]$, and then process the new vector sequence with a RNN with hidden state $\mathbf{s}_k^t \in \mathbb{R}^S$, whose update equation reads $\mathbf{s}_j^t = g_\theta(\tilde{\mathbf{w}}_j, \mathbf{s}_{j-1}^t)$. Here g_θ is implemented by a LSTM network, with equations similar to those below Eq. (2).

The distribution p_θ is then defined as a categorical distribution over a vocabulary of size V, whose class probabilities are given by $\boldsymbol{\pi}_j^t = \text{softmax}(\mathbf{W}\,\mathbf{s}_j^t)$, where $\mathbf{W} \in \mathbb{R}^{V \times S}$ is a learnable matrix.

We now define the summary representation for $\mathbf{x}_t$ as a weighted sum over word representations $\bar{\mathbf{s}}_t = \sum_j^{L_t} \alpha_j^t \mathbf{s}_j^t$ where the j-th weight α_j^t is calculated with the *gated attention mechanism* proposed in [24]

$$\alpha_j^t = \text{softmax}(\mathbf{k}_j^\top \mathbf{q}),$$
$$\mathbf{k}_j = \tanh(\mathbf{M}_1 \mathbf{s}_j^t + \mathbf{b}_1) \odot \sigma(\mathbf{M}_2 \mathbf{s}_j^t + \mathbf{b}_2), \tag{6}$$

where $\mathbf{M}_1, \mathbf{M}_2 \in \mathbb{R}^{A \times S}$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^A$ are learnable parameters, $\mathbf{q} \in \mathbb{R}^A$ can be interpreted as a learnable global query, $\odot$ denotes element-wise multiplication and $\sigma(\cdot)$ denotes the sigmoid function. This type of attention is introduced to solve the problem of the limited expressiveness of the $\tanh(\cdot)$ to capture complex relations, due to the fact of approximate linearity in the region $[-1, 1]$.

As we shall see below, this attentive summary representation allows us to track the most relevant words affecting the rating of a given item as time evolves.

C. Combining temporal and summary representations

Given the temporal representations for user and item reviews (i.e. $\mathbf{h}_t^u$, $\mathbf{h}_t^v$), and the summary representation for review content $\bar{\mathbf{s}}_t$, we want to predict the rating $\hat{y}_t^{uv} \in \mathbb{R}$ that user u gives to item v. There is, however, still the question of how to combine $\mathbf{h}_t^u$ and $\mathbf{h}_t^v$ with $\bar{\mathbf{s}}_t$. After exploring different possibilities we found two optimal solutions namely,

1) DRR-BoW: For the Bow Neural Review Model we augment Eq. (1) and define $\tilde{\mathbf{z}}_t^e = \mathbf{z}_t^e + \mathbf{W}_s^e \bar{\mathbf{s}}_t$, where $\mathbf{W}_s^e \in \mathbb{R}^{H \times S}$ is an additional learnable weight, to get $\tilde{\mathbf{h}}_t^e = f_\theta^{(e)}(\tilde{\mathbf{z}}_t^e, \tilde{\mathbf{h}}_{t-1}^e)$, where $f_\theta^{(e)}$ remains the same as in Eq. (2). The new representation $\tilde{\mathbf{h}}_t^e$ now encodes the nonlinear interaction between timing, rating *and text* of past reviews.

2) DRR-LM: For the Autoregressive Review Model we instead define

$$\tilde{\mathbf{h}}_t^e = \mathbf{W}^{(e)}\text{concat}([\mathbf{h}_t^e, \bar{\mathbf{s}}_t]) + \mathbf{b}^{(e)}, \tag{7}$$

with $\mathbf{W}^{(e)} \in \mathbb{R}^{H \times (H+S)}, \mathbf{b}^{(e)} \in \mathbb{R}^H$ learnable. The resulting representation $\tilde{\mathbf{h}}_t^e$ also encodes the interaction between timing, rating and text, albeit through a different route.

D. Rating Model

We have now all ingredient to predict the rating $\hat{y}_t^{uv} \in \mathbb{R}$ that user u gives to item v. We compute $\hat{y}_t^{uv}$ with a factorization machine (FM) [25], here defined as

$$\hat{y}_{t+1}^{uv}(\mathbf{h}) = w_0 + \sum_{i=1}^{2H} w_i h_i + \sum_{i=1}^{2H} \sum_{j=i+1}^{2H} \langle \mathbf{v}_i, \mathbf{v}_j \rangle h_i h_j, \tag{8}$$

where $\mathbf{h} \in \mathbb{R}^{2H} \equiv \text{concat}([\tilde{\mathbf{h}}_t^u, \tilde{\mathbf{h}}_t^v])$, $w_0 \in \mathbb{R}$, $\mathbf{w} \in \mathbb{R}^{2H}$ and $\mathbf{V} \in \mathbb{R}^{2H \times K}$ are learnable parameters, K is set to 10 and $\langle \cdot, \cdot \rangle$ denotes dot product.

We choose the loss function of the Rating Model to be the mean square error function between $\hat{y}_t^{uv}$ and our prediction $\hat{y}_t^{uv}(\mathbf{h})$.

E. DRR Loss function

The complete loss function of the DRR model has therefore three components: the loss of the Rating Model, the loss of the Dynamic Model of Review Sequences, which is the negative log-likelihood of an exponential, and the loss of the Dynamical Model of Review Content, which is the negative log-likelihood of our word sequence model. Explicitly we write

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{u,v \in \mathcal{D}} \sum_t (y_t^{uv} - \hat{y}_{t=1}^{uv})^2$$
$$- \lambda_1 \sum_{e \in \mathcal{D}} \sum_{t=1}^{N_e} \log p_\theta(\delta_{t+1}^e | \mathbf{h}_t^e) - \lambda_2 \sum_t \log p_\theta(\mathbf{x}_t | \mathbf{h}_{t-1}),$$
(9)

where $\lambda_1, \lambda_2 \in \mathbb{R}^+$ are hyperparameters.

IV. CAUSALITY

By construction, both DRR-BoW and DRR-LM models above preserve causality — the models do not use *any* information from the future to predict ratings. As mentioned in the introduction, however, most recommender system models that leverage review content use the review $\mathbf{x}_{t+1}^v$, written by user u, to predict the rating y_{t+1}^{uv} given by this same user to the item v. In order to fairly compare our methodology with such models, we use the degrees of freedom available within the definition of the DRR-LM model and redefine

$$\tilde{\mathbf{h}}_t^e = \mathbf{W}^{(e)} \text{concat}([\mathbf{h}_t^e, \bar{\mathbf{s}}_{t+1}]) + \mathbf{b}^{(e)}. \qquad (10)$$

This new representations encodes $\bar{\mathbf{s}}_{t+1}$, the summary representation of the review whose rating it predicts, and breaks causality. Below we refer to the model using the causal representation Eq. (7) as DRR-LM-C, whereas we denote the model using the non-causal expression Eq. (10) as DRR-LM-NC.

Naturally, the causal model is to be preferred as we normally do not have review content about the item whose rating we want to predict. Nevertheless, we shall see that the non-causal model lends itself when one is interested in tracking the words which most affect the rating of a given item as time evolves.

V. EXPERIMENTS AND RESULTS

Data set To test our model we choose the Amazon dataset [26]. We pick four 5-core subcategory datasets namely, *Automotive (A)*, *Digital Music (DM)*, *Tools and Home (TH)* and *Pet Supplies (PS)*. The review creation time is defined as the difference in days between the original timestamp and the timestamp of the first review in the dataset. Next we group reviews by day, since the granularity of the timestamps is day based. All users or items with less than 5 days (i.e. time series with less than 5 points) are removed from the dataset. The autoregressive language models use the review raw text, changed into lower case. Preprocessing scripts can be found at [1]. Statistics of the preprocessed data is summarized in Table I.

Training Our model predicts ratings through the user and item dynamic representations, which come from two independent RNNs. Simply applying backpropagation through both sequences is computationally forbidden. In order to overcome this problem, we train the user and item RNNs alternately. We first freeze the parameters of e.g. the items' RNN, and only update those of the users' RNN, while back-propagating the gradients of all ratings for a user batch. The items' dynamic representations are taken to be fixed. We then repeat these operations but now with the user parameters and user representations frozen.

Model Configuration We split each dataset along the time dimension into three parts: training set (80%), validation set (10%) and test set (10%). We use grid search on the validation set for hyperparameter tuning. We set the hidden dimension H of the temporal representation $\mathbf{h}_t^e$ to 32, and the embedding dimension E of $\mathbf{z}_t^e$ to 100. Regarding the review content models, we set the vocabulary size V to 2000 for DRR-BoW and to 5000 for DRR-LM. In the latter case we also use GloVe word embeddings [27] (these corresponds to the $\mathbf{w}_j^t$ in Eq. (5)) with dimension 300. For DRR-LM we also set the attention dimension A to 64 and the embedding dimension H' of the (concatenation of the) temporal and summary representations to 64. We use Adam [28] with learning rate 0.0002 and $\beta_1 = 0.9$ and limit the review length to 150 tokens. All methods are implemented using PyTorch v1.3[1]. Source code for all models can be found at [1].

Results Given an user and item of interest, the DRR model predicts the arrival time, rating and the probability over the word sequence of the next review, and we optimize the model to give the best performance on the rating prediction task.

Our methodology incorporates modeling the dynamic aspects of user-item interaction with neural models of review content. To test the importance of each of these components for the problem of rating prediction, we test our models against (i) the Probabilistic Matrix Factorization (PMF) [29], which is a static recommender system which does not model review content; (ii) the RRN [10], a causal model which learns dynamic user/item representations (albeit non-continous), but does not model review content; and (iii) three static models which do leverage review content, namely DeepCoNN [14], D-ATT [13] and AHN [12]. These last three models are non-casual since they either use the review of the item whose rate they predict, or use item reviews that have not been received by the time the item of interest was rated. Table II shows results for all models on the chosen datasets. We use boldface to highlight best results in both causal and non-causal cases.

Let us start by focusing on the causal models. First we note that both DRR-BoW and DRR-LM-C outperform the RRN model, which confirms the known fact that review content helps in rating prediction tasks. We remark however that in this case the models in questions are dynamic, and it is the content of *past reviews* what is successfully being used. Interestingly, DRR-BoW beats DRR-LM-C which may hint at the fact that
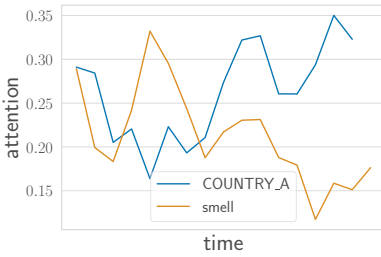
---

[1]https://pytorch.org/

TABLE I: Datasets statistics. The mean and the standard deviation of the number of reviews, sentences and words per review with respect to the user and item.

	Automotive		Digital Music		Tools and Home		Pet Supplies	
	user/item		user/item		user/item		user/item	
	mean	std	mean	std	mean	std	mean	std
reviews	6.1/9.3	1.7/5.5	7.8/9.0	6.2/6.5	7.8/10.6	5.3/9.0	7.44/13.7	4.4/14.2
sentences	8.7/9.7	6.3/7.5	6.3/4.9	11.9/10.2	8.4/7.5	8.1/7.8	7.8/6.5	7.7/6.9
words	89.8/101.6	68.3/82.5	52.5/38.2	106.4/94.6	80.3/70.5	85.6/83.3	68.9/55.9	71.1/66.2

TABLE II: Mean-square error on the rating prediction (* results taken from [12]).

	static	non-causal models				causal-models		
Datasets	PMF*	DeepCoNN*	D-ATT*	AHN*	DRR-LM-NC	RRN	DRR-BoW	DRR-LM-C
A	0.9187	0.7809	0.7654	**0.7314**	0.7791	1.0927	**0.7838**	0.8171
DM	0.8788	0.8754	0.8506	0.8172	**0.7250**	0.7961	**0.7723**	0.7801
TH	1.1182	0.9856	0.9850	0.9671	**0.9264**	1.0896	**1.0406**	1.0656
PS	1.4340	1.2598	1.2730	1.2515	**1.0500**	1.1970	**1.1734**	1.1918

Fig. 2: *Upper Left:* Dynamic attention on the words 'COUNTRY_A' and 'smell' for an item in 'Pet Supplies' dataset. *Upper Middle:* Review sample from the beginning of the time series. *Upper Right:* Review sample from the end of the time series. **The real names of the countries are replaced with masks 'COUNTRY_A' and 'COUNTRY_B' for fairness. *Lower Left:* Dynamic attention on the words 'comfortable' and 'ear' for an item in the 'Tools and Home' dataset. *Lower Middle:* Review sample from the beginning of the time series. *Lower Right:* Review sample from the end of the time series. The darker the highlight color for a word, the higher its attention value.

it is enough to know that certain key words are present in the review, as opposite to e.g. word order, to better predict the rating. Regarding the non-causal models, DRR-LM-NC outperforms all other models in almost all datasets, which shows that one indeed needs to not only account for review content, but also for its dynamic character. Remarkably, both causal models DRR-BoW and DRR-LM-C perform better than all their non-causal competitors in two of the datasets (see the Digital Music and Pet Suplies rows in the table), and comparable to them in the others.

We can conclude that our models successfully learn both temporal user/item representations and review content representation which *together* are useful for rating prediction.

Let us now consider the dynamic attention mechanism of

the DRR-LM-NC, which allows us to e.g. follow in time the weights α_j^t (defined in Eq. 6) of the words in the reviews for the item whose rate we aim at predicting. The higher the weight of a word, the stronger its relevance to the rating prediction. Figure 2 *Upper Left* shows the attention weights on the words 'COUNTRY_A' and 'smell' as time evolves for a given product in the 'Pet Supplies' dataset. One can see that although at the start of the time series the word 'smell' was important for determining the rating, its relevance decreases as the weight on the word 'COUNTRY_A' increases. After a closer look at the reviews we learn that at the start of the time series most reviews were related to the smell of the product (e.g. whether the dogs were liking the product's smell). Later on, however, the manufacturing company moved the product

production to COUNTRY_A, and this event was successfully captured by our attention model. Figure 2 *Upper Middle* shows an example review for the item in question, from the start of the time series. Words with darker highlights mean here words with higher attention weight. One can see that the word 'smell' is highlighted as important. In contrast, Figure 2 *Upper Right* displays a review sampled from the end of the time series, in which one sees the word 'COUNTRY_A' has more relevance than the word 'smell'. Similarly, the *Lower* row of Figure 2 shows the attention weights on the words 'comfortable' and 'ear' as time evolves for a given product in the 'Tools and Home' dataset.

VI. Conclusion and Feature Work

In this work we proposed a recommender system model which accounts for the dynamic aspects of user preferences, as reflected in their history of reviews and ratings. We explicitly learn continuous-time representations for both users and items, and use these to define dynamic language models for review content. The latter provided us with review content representations which, when combined with the temporal user/item representations, proved to be useful in predicting the hidden interest of users in unknown items. Indeed, our results outperformed several state-of-the-art recommender system models in rating prediction tasks, in different datasets.

We also introduced a new dynamic attention mechanism which allowed us to track the most relevant words for a given rating of an item of interest at a given instant of time.

Future directions of work include developing attention mechanisms between reviews with different timestamps, and learning more generic dynamic representations able to characterize hidden dynamics global to all users.

Acknowledgment

References

[1] "Source code," https://figshare.com/s/2e19d38501d275944487.

[2] Y. Koren, "Collaborative filtering with temporal dynamics," *Commun. ACM*, vol. 53, no. 4, p. 89–97, Apr. 2010. [Online]. Available: https://doi.org/10.1145/1721654.1721677

[3] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[4] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 791–798.

[5] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 111–112.

[6] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[7] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 130–137.

[8] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016, pp. 17–22.

[9] B. Twardowski, "Modelling contextual information in session-aware recommender systems with neural networks," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 273–276.

[10] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 495–503.

[11] R. Catherine and W. Cohen, "Transnets: Learning to transform for recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 2017, pp. 288–296.

[12] X. Dong, J. Ni, W. Cheng, Z. Chen, B. Zong, D. Song, Y. Liu, H. Chen, and G. de Melo, "Asymmetrical hierarchical networks with attentive interactions for interpretable review-based recommendation," *ArXiv*, vol. abs/2001.04346, 2019.

[13] S. Seo, J. Huang, H. Yang, and Y. Liu, "Interpretable convolutional neural networks with dual local and global attention for review rating prediction," in *Proceedings of the eleventh ACM conference on recommender systems*, 2017, pp. 297–305.

[14] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 425–434.

[15] C.-Y. Wu, A. Ahmed, A. Beutel, and A. J. Smola, "Joint training of ratings and reviews with recurrent recommender networks," 2016.

[16] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg, "Feature hashing for large scale multitask learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 1113–1120.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[18] K. Cvejoski, R. J. Sanchez, B. Georgiev, J. Schuecker, C. Bauckhage, and C. Ojeda, "Recurrent point processes for dynamic review models," in *Workshop on Interactive and Conversational Recommendation Systems at AAAI*, 2020.

[19] K. Cvejoski, R. J. Sánchez, B. Georgiev, C. Bauckhage, and C. Ojeda, "Recurrent point review models," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[20] Y. Miao, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *International conference on machine learning*, 2016, pp. 1727–1736.

[21] G. E. Hinton and R. R. Salakhutdinov, "Replicated softmax: an undirected topic model," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 1607–1614. [Online]. Available: http://papers.nips.cc/paper/3856-replicated-softmax-an-undirected-topic-model.pdf

[22] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model." in *INTERSPEECH*, T. Kobayashi, K. Hirose, and S. Nakamura, Eds. ISCA, 2010, pp. 1045–1048.

[23] S. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 10–21.

[24] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," *arXiv preprint arXiv:1802.04712*, 2018.

[25] S. Rendle, "Factorization machines," in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 995–1000.

[26] R. He and J. McAuley, "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*, 2016, pp. 507–517.

[27] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2008, pp. 1257–1264.