RARE : A Recurrent Attentive Recommendation Engine for News Aggregators

Dhruv Khattar
International Institute of Information Technology Hyderabad, India
dhruv.khattar@research.iiit.ac.in

Vaibhav Kumar*
International Institute of Information Technology Hyderabad, India
vaibhav.kumar@research.iiit.ac.in

Shashank Gupta
International Institute of Information Technology Hyderabad, India
shashank.gupta@research.iiit.ac.in

Manish Gupta†
International Institute of Information Technology Hyderabad, India
manish.gupta@iiit.ac.in

Vasudeva Varma
International Institute of Information Technology Hyderabad, India
vv@iiit.ac.in

Abstract

With news stories coming from a variety of sources, it is crucial for news aggregators to present interesting articles to the user to maximize their engagement. This creates the need for a news recommendation system which understands the content of the articles as well as accounts for the users' preferences. Methods such as Collaborative Filtering, which are well known for general recommendations, are not suitable for news because of the short life span of articles and because of the large number of articles published each day. Apart from this, such methods do not harness the information present in the sequence in which the articles are read by the user and hence are unable to account for the specific and generic interests of the user which may keep changing with time. In order to address these issues for news recommendation, we propose the Recurrent Attentive Recommendation Engine (RARE).

RARE consists of two components and utilizes the distributed representations of news articles. The first component is used to model the user's sequential behaviour of news reading in order to understand her general interests, i.e., to get a summary of her interests. The second component utilizes an article level attention mechanism to understand her specific interests. We feed the information obtained from both the components to a Siamese Network in order to make predictions which pertain to the user's generic as well as specific interests. We carry out extensive experiments over three real-world datasets and show that RARE outperforms the state-of-the-art. Furthermore, we also demonstrate the effectiveness of our method in handling the cold start cases.

1 Introduction

A news aggregator collects news from a variety of sources and presents it to the user. It would be quite cumbersome for a user to select articles of her choice from a huge list of presented articles which may pertain to a variety of subjects. Hence, it becomes crucial for such aggregators to have a recommendation system to point the user to the most relevant items and thus maximize her engagement with the site and minimize the time needed to find relevant content.

A popular approach to the task of recommendation is collaborative filtering [2, 20, 23], which uses the user's past interaction with the item to predict the most relevant content. Another common approach is content-based recommendations, which uses features between items and/or users to recommend new items to the users based on the similarity between features. However, amongst the various approaches for collaborative filtering, Matrix Factorization (MF) [14], is the most popular one, which projects users and items into a shared latent space, using a vector of latent features to represent a user or an item. Thereafter, a user's interaction with an item is modelled as the inner product of their latent vectors.

However, Collaborative Filtering methods are not suitable for news recommendation because news articles have a short life span and expire quickly [30]. Such methods also require a considerable number of interactions with an item (article) before making predictions which is not desirable for news recommendation because we would ideally want to start recommending articles as soon as they are published. Also, they do not directly harness the information present in the sequence in which the articles were read by the user and hence fail to account for the generic as well as specific interests of the user which may keep changing with time. In order to address these issues it becomes crucial to understand the content of the news articles as well as the user's preferences. We explain this through an example in the following paragraph.

As can be seen from Fig. 1(A), if a user reads four different articles belonging to tennis and football, then we would like our model to infer that the generic interests of the users lie in reading articles about sports. Hence, this would allow articles belonging to different topics in the sports category to be recommended to the user. However, since the user reads more articles on tennis rather than football, we would like to give more weight to the articles related to tennis as can be seen in Fig. 1(B). Hence, in our overall list of recommended articles to the user, we would like to present news articles related to sports amongst which articles related to tennis would be given more importance. It may also happen that the user suddenly starts reading articles related to business rather than sports. In such a case we may also want to

*Author had equal contribution. He can also be contacted at vaibhav2@andrew.cmu.edu
†Author is also a Principal Applied Researcher at Microsoft.

INRA'18 in conjunction with CIKM'18, October 26, 2018, Torino, Italy

start recommending articles related to business as well. This can be seen in Fig. 1(C). However, it is important to note that in all these cases the sequential reading history of the user is very important while generating recommendations.

To encode this intuition, we propose a novel neural network framework namely Recurrent Attentive Recommendation Engine (RARE). As illustrated in Fig. 3, RARE consists of two components. The first component is based on a recurrent neural network and uses the sequential reading history of the user as its input. We call this the generic encoder. This helps us to identify the generic/overall interests of the users, i.e., it provides a summary of the user's interests. The second component utilizes a recurrent neural network with an attention mechanism to identify the specific interests of the user. We call this the specific encoder. The part dealing with attention allows the model to attend to articles in a differential manner, discriminating the more from the less important ones. We then concatenate the representations obtained from both these components and call it the unified representation of the users' interests. Limiting the size of the user reading history used as inputs to both these components allows us to adapt to the changing user preferences. We then feed this unified representation along with the representation of the candidate article to a Siamese Network and compute an element wise product between the outputs obtained at the final layer of the sister networks, as illustrated in Fig. 2. Finally, we use a logistic unit to compute the score for recommendation. Using such a network enhances the model with further non-linearity and enables it to capture the user-article interaction in a better sense. It also allows the model to learn an arbitrary similarity function instead of the traditional metrics. The distributed representation of each news article is used as input to our model. This gives us the capability to recommend articles as and when they are produced, without depending on any prior user interaction with that article.

To summarize, the main contributions of this work are as follows.

- We present a neural network based architecture (RARE) with the following capabilities.
  - It utilizes the content of the news articles giving it the ability to recommend articles as soon as they are published.
  - It takes into account the users' generic as well as specific interests.
  - It adapts to the changing interests of the user.
- We carry out extensive experiments over three real world datasets to show the effectiveness of our model. The results reveal that our method outperforms the state-of-the-art.
- We show the effectiveness of our model for solving the cold-start cases as well.

2 Related Work

There has been extensive study on recommendation systems with a myriad of publications. In this section, we aim at reviewing a representative set of approaches.

2.1 Common Approaches for Recommendation Systems

Recommendation systems in general can be divided into collaborative recommendation systems and content-based recommendation systems. In collaborative filtering based recommendations,

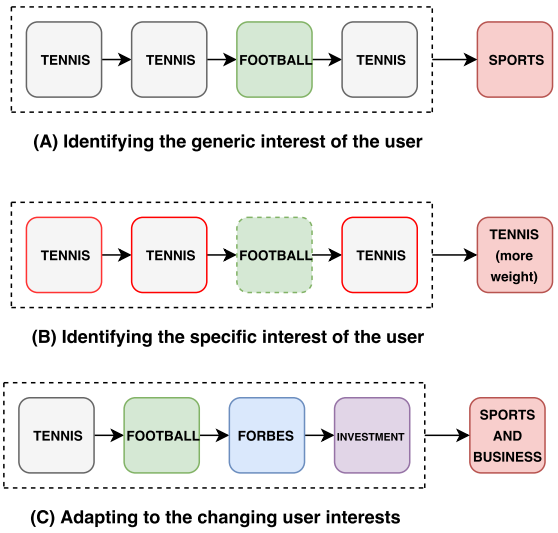

Figure 1: In (A), the user's sequence is used to model her general interests. While in (B), the user's specific interests are captured. In (C), the changing interests of the user are modelled. In all these cases, sequential reading history of a user plays an important role. Different colors represent the different topics of the article.

an item is recommended to a user if similar users liked that item. Collaborative filtering can be further divided into user collaborative filtering, item collaborative filtering or a hybrid of both user and item collaborative filtering. Examples of such techniques include Bayesian matrix factorization [22], matrix completion [20], Restricted Boltzmann Machine [23], nearest neighbour modelling [2]. In user collaborative methods such as [2], the algorithm first computes similarity between every pair of users based on the items liked by them. Then, the scores of user-item pairs are computed by combining scores of this item given by similar users. Item-based collaborative filtering [24], computes similarity between items based on the users who like both items. It then recommends items to the user based on the items she has previously liked. Finally, in user-item based collaborative filtering, both the users and the items are projected into a common vector space based on the user-item matrix and then the item and user representation are combined to find a recommendation. Matrix factorization based approaches like [20] and [22] are examples of such a technique. One of the major drawbacks of collaborative filtering is its inability to handle new users and new items, a problem which is often referred to as the cold-start issue.

Another common approach for recommendation is content-based recommendation. In this approach, features from user's profile and/or item's description are extracted and are used for recommending items to users. The underlying assumption is that the users tend to like items that they liked previously. In [16], each user is modeled by a distribution over news topics that is constructed from articles she liked with a prior distribution of topic preferences computed using all users who share the same location. A major advantage of using content-based recommendation is that it can handle the problem of item cold-start as it uses item features for

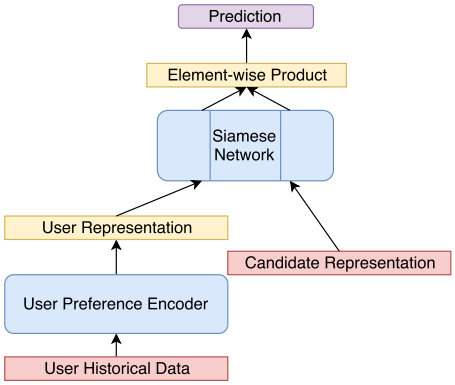

Figure 2: RARE Model Architecture

recommendation. For user cold-start, a variety of other features like age, location, popularity aspects could be used. In the following we discuss previous work on neural approaches for recommendation systems.

2.2 Neural Recommendation Systems

Early work which used neural networks [23] used a two-layer Restricted Boltzmann Machine (RBM) to model users' explicit ratings on items. The work has been later extended to model the ordinal nature of ratings [18]. Recently auto-encoders have become a popular choice for building recommendation systems [3, 25, 26]. The idea of user-based AutoRec [25] is to learn hidden structures that can reconstruct a user's ratings given her historical ratings as inputs. In terms of user personalization, this approach shares a similar spirit as the item-item model [17, 24] that represents a user in terms of her rated item features. While previous work has lent support for addressing collaborative filtering, most of them have focused on observed ratings and modeled the observed data only. As a result, they can easily fail to learn users' preferences from the positive-only implicit data.

In [28] a collaborative denoising auto-encoder (CDAE) for CF with implicit feedback is presented. In contrast to the DAE-based CF [26], CDAE additionally plugs a user node to the input of auto-encoders for reconstructing the user's ratings. As shown by the authors, CDAE is equivalent to the SVD++ model [14] when the identity function is applied to activate the hidden layers of CDAE. Although CDAE is a collaborative filtering model, it is solely based on item-item interaction whereas the work which we present here is based on user-item interaction. On the other hand in [9], authors have explored deep neural networks for recommendation systems. They present a general framework named NCF, short for Neural Collaborative Filtering, that replaces the inner product with a neural architecture that can learn an arbitrary function from the given data. It uses a multi-layer perceptron to learn the user-item interaction function. NCF is able to express and generalize matrix factorization. They then combine the linearity of matrix factorization and non-linearity of deep neural networks for modelling user-item latent structures. They call this model as NeuMF, short for Neural Matrix Factorization.

Since our work also involves projecting articles and users to a common geometric space, we review the work in [13]. They propose an effective approach for projecting queries and documents into a common low-dimensional space. The model is named as Deep Structured Semantic Model (DSSM) [13] and is effective in calculating the relevance of the document given a query by computing the distance between them. Originally this model was meant for the purpose of ranking, but since the problem of ranking has very close associations with that of recommendation, DSSM was later extended to recommendation scenarios in [6]. In [6], the authors designed a DSSM such that the first neural network contains user's query history (and thus referred to as the user view) and the second neural network contains implicit feedback of items. The resulting model is named multi-view DNN (MV-DNN) since it can incorporate item information from more than one domain and then jointly optimize all of them using the same loss function in DSSM. However, in [6], the features for the users were their search queries and features for items came from multiple sources (e.g., Apps, Movies/TV etc.). This makes it less adaptable by a news website as it requires a lot of information outside the news domain. However, if the work is viewed in its entirety, it suggests that supercharging a neural network with non-linearities to project a user and an item to the same geometric space is very effective in calculating relevance. We draw the inspiration for using Siamese network in our model on similar grounds.

3 Model Architecture

In this section we first introduce the news article recommendation task and then provide an elaborate description of the various components of the proposed RARE model.

3.1 Task Description

Given a series of news articles read by the user, our task is to recommend articles of interest to the user. The implicit feedback provided by the user is available to us, i.e., we have information about the articles clicked by the user. Apart from this, we also have the content of the news articles available at our disposal.

We first select a reading history of size R for each user. The size of the reading history determines the number of past interactions we use for making predictions. The articles previously read by a user can be represented as $[r_1, r_2, ..., r_t, ..., r_R]$ where $1 \leq t \leq R$. Using this list as inputs to our model we need to recommend a ranked list of articles which are aligned with the users' interests.

3.2 RARE Overview

We propose a novel Recurrent Attentive Recommendation Engine (RARE) to address the problem of news recommendation for news aggregators. An overview of our method can be seen in Fig. 2. The basic idea of RARE is to build a unified representation of a user's interests which encapsulates both her specific and generic interests. Apart from this, using a specific amount of reading history of a user provides RARE with the flexibility to adapt to the changing interests of the user. The pipeline of RARE can be described as follows.

- We first learn a distributed representation for each news article by combining its title and text.

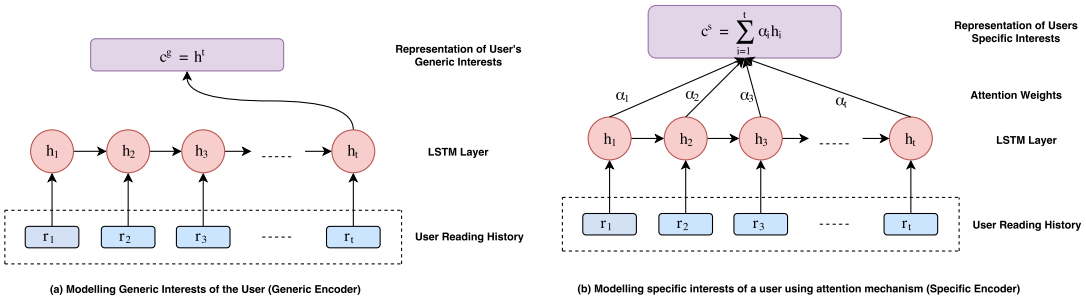

Figure 3: Two Components of RARE: Generic Encoder and Specific Encoder

- We then fix a reading history size R, and use the representations of the previous R articles read by the user as inputs to the model.
- We come up with a unified representation of the users' interests using recurrent neural networks with an attention mechanism.
- Treating the unified representation of the user as a query and the representation of the candidate article as a document, we use a Siamese network to make them undergo similar transformations and supercharge them with non-linearities to discover user-item interactions.

3.3 Distributed Representation for News Articles

We learn a 300-dimension distributed representation [15] for each news article by combining the title and text of the news articles. Learning such a representation allows us to

- Capture the overall semantics of the news article.
- Enables the model to come up with a representation for new news articles as well as of articles with varying lengths.

News articles generally follow an inverted pyramid structure where the title and the first paragraph give away the desired information. Hence, we only choose the title and the first paragraph because it usually contains all the relevant information without delving into detailed explanations. We also experimented by choosing the entire news article but found better results with just the first paragraph.

3.4 Generic Encoder

The inputs for the generic encoder are the representations of the articles previously read by the user. Fig. 3(a) shows the graphical model of the network used to identify generic interests in RARE. We use Recurrent Neural Network (RNN) with Long-Short Term Memory (LSTM) cells. LSTMs have been shown to be capable of learning long-term dependencies [11, 27]. The aim of this component is to understand the generic (broader/overall) interests of the user. The last hidden state of the RNN, i.e., h_t encapsulates this information, which we represent as c^g. We can think of the final hidden state as the overall summary of the user's interests.

The state updates of the LSTM satisfy the following equations.

$$f_t = \sigma\left[W_f\left[h_{t-1}, r_t\right] + b_f\right] \tag{1}$$

$$i_t = \sigma\left[W_i\left[h_{t-1}, r_t\right] + b_i\right] \tag{2}$$

$$o_t = \sigma\left[W_o\left[h_{t-1}, r_t\right] + b_o\right] \tag{3}$$

$$l_t = \tanh\left[V\left[h_{t-1}, r_t\right] + d\right] \tag{4}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot l_t \tag{5}$$

$$h_t = o_t \cdot \tanh(c_t) \tag{6}$$

Here σ is the logistic sigmoid function. f_t, i_t, o_t represent the forget, input and output gates respectively. r_t denotes the input at time t and h_t denotes the latent state. W_f, W_i, W_o and V represent the weight parameters respectively, while b_f, b_i, b_o and d represent the bias parameters respectively. The forget, input and output gates control the flow of information throughout the sequence.

3.5 Specific Encoder

The architecture of Specific Encoder is similar to that of the Generic Encoder. The graphical representation for this can be seen in Fig. 3(b). We use LSTM cells here as well. To capture the specific interests of the users, i.e., to understand the deeper interests of the user within her broader interests, we use an article level attention mechanism. This provides us with a context vector which encapsulates the specific interests of the user. This can be represented as,

$$c^s = \sum_{j=1}^{R} \alpha_j h_j \tag{7}$$

where the attention weights, α_j, control the part of the input sequence which should be emphasized or ignored and h_j stands for the output of the hidden units. This attention mechanism gives RARE the capability to adaptively focus more on the important items.

3.6 RARE

The complete architecture of the proposed model can be seen in Fig. 4. The outputs obtained from the specific and the generic encoder are concatenated and then used as inputs to a Siamese network along with the candidate article.

For the given task, the generic encoder captures the overall interests of the user, i.e., it captures the summary of the entire news articles read by the user. At the same time, the specific encoder adaptively selects the important articles to capture the specific interests of the user. Hence to take advantage of both kinds of information we concatenate the outputs of both the encoders.

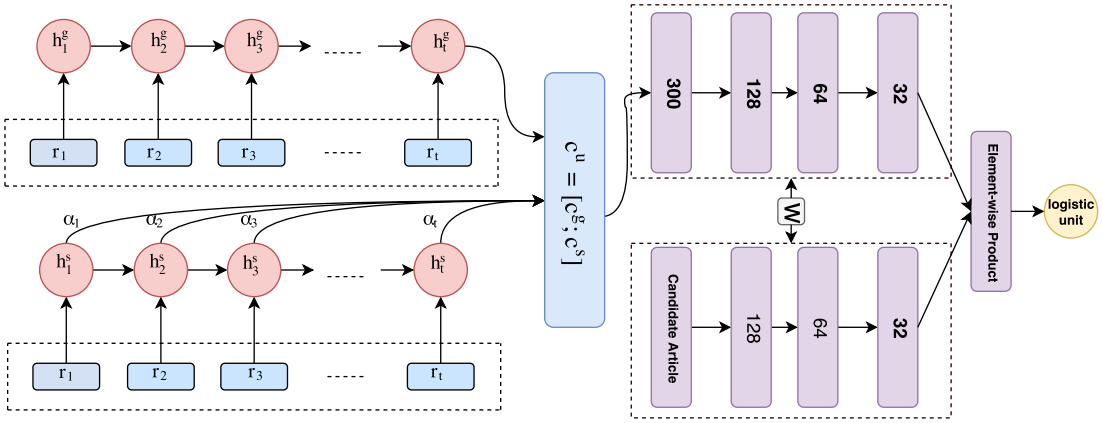

Figure 4: Complete Architecture of the RARE System

As shown in Fig. 3, we can see that h_t^g is incorporated into c_u to provide the summarized user interests. Note that different encoding mechanisms will be invoked in both the encoders when trained jointly. The last hidden state of the generic encoder h_t^g plays a different role from that of h_t^s. The former has the responsibility to encode the information present in the sequence in which the articles were read by the user. While the latter is used for computing attention weights. Information obtained from both the encoders is utilized to come up with a unified representation of users' interests.

$$c^u = [c^g; c^s] = [h_t^g; \sum_{j=1}^{R} \alpha_j h_j^s] \tag{8}$$

where c^u represents the unified representation of users' interests.

We then use c^u as inputs to one of the sister networks in the Siamese network as shown in Fig. 3. The input to the other sister network is the learned representation of the candidate article. The Siamese network supercharges RARE with further non-linearities and makes the user representation and the article representation go through similar transformations. In [13], an architecture similar to that of a Siamese network has been used for ranking documents with respect to a query with great effectiveness. If we try to draw a parallel between the query-document problem with our task, one can see that a query in our case is c_u and the document is the representation of the candidate news article. Hence, it seems apt to use such a network if we were to project both of these into the same geometric space to uncover the underlying user-article interaction pattern. A similar sort of technique has also been used by authors in [9] for modelling user-item interactions. Final predictions are obtained from the Siamese network after the logistic on the element-wise product between the outputs obtained from the sister networks.

Rather than using Siamese networks, the other choice was to use a typical encoder-decoder framework. However, a typical encoder-decoder framework is unable to produce out-of-vocabulary (OOV) words. In the news recommendation problem setting, each new published article, that has not been interacted by any user would act as an "OOV word". However, it is very crucial for a news recommender to recommend articles as soon they are published which is why we resort to such a method as it allows us to handle such cases well.

3.7 Learning

Typically, to learn the model parameters, existing point-wise methods [21] perform regression with a squared loss. This is based on the assumption that observations are generated from a Gaussian distribution. However, in [9] it has been shown that such a method is not very effective when we have implicit data available.

Given a user u and an article x, let $\hat{y_{ux}}$ represent the predicted score at the output layer. Training is performed by minimizing the point-wise loss between $\hat{y_{ux}}$ and its target value y_{ux}. Considering the one-class nature of implicit feedback, we can view the value of y_{ux} as a label 1 meaning the item x is relevant to a user u, and 0 otherwise. The prediction score $\hat{y_{ux}}$ then represents how likely an item x is relevant to u. Hence in order to constrain the values between 0 and 1, we use the logistic function. We then define the likelihood function as follows.

$$p(\gamma^+, \gamma^- | I, \Theta_m) = \prod_{(u,i) \in \gamma^+} \hat{y_{ui}} \prod_{(u,j) \in \gamma^-} (1 - \hat{y_{uj}}) \tag{9}$$

where γ^+ and γ^- represent the positive (observed interactions) and negative (unobserved interactions) articles respectively. I represents the input and Θ_m represents the parameters of the model. The negative log likelihood can then be written as follows (after rearranging the terms).

$$L = - \sum_{u,i \in \gamma^+ \cup \gamma^-} y_{ui} \log \hat{y_{ui}} + (1 - y_{ui})(1 - \log \hat{y_{ui}}) \tag{10}$$

The loss is similar to binary cross-entropy and can be minimized using gradient descent methods.

It is also worth noticing that the likelihood function is such that it simultaneously adjusts the model's parameters by maximizing the score of the relevant articles and at the same time adjusts to minimize the score of the non-relevant articles. This is similar to what is done while ranking documents corresponding to a query in [13]. Using such a likelihood also gives us the advantages of a ranking function.

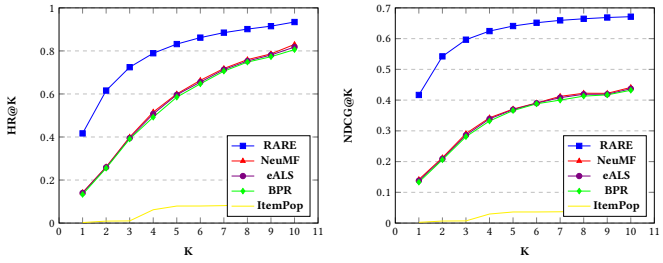

Figure 5: Performance of RARE vs state-of-the-art on CLEF NewsREEL

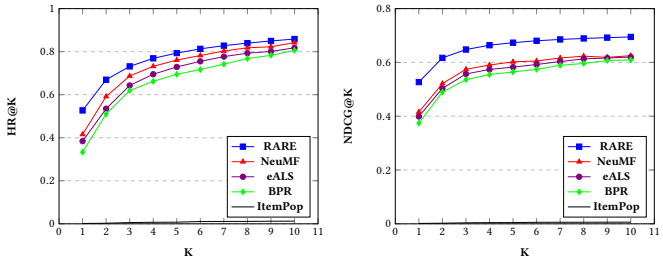

Figure 6: Performance of RARE vs state-of-the-art on Malayalam Dataset

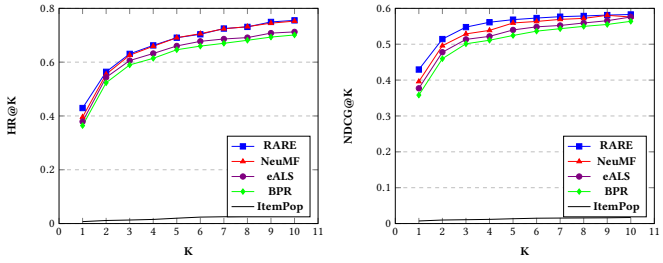

Figure 7: Performance of RARE vs state-of-the-art on Indonesian Dataset

4 Experiments

In this section, we describe the datasets, the state-of-the-art methods, evaluation protocol along with the settings used for learning the parameters of the model.

4.1 Dataset

We use three real world datasets for evaluation. First, we use the dataset published by CLEF NewsREEL 2017 [12]. CLEF shared a dataset which captures interactions between users and news stories. It includes interactions of eight different publishing sites in the month of February 2016. The recorded stream of events include 2 million notifications, 58 thousand item updates, and 168 million recommendation requests. It also includes information like the title and text of each news article. For this dataset we considered all the users who had read more than 10 articles after which we get a total of 22229 users. The other two datasets are provided by a popular news aggregation website (name omitted for review). The second dataset contains a list of articles read by 10297 users in an Indian language, Malayalam. The third dataset contains a list of articles read by 22848 users in Indonesian. We make the code publicly available [1].

4.2 Baselines

We compare our proposed approach with the following methods.

- **ItemPop**. News articles are ranked by their popularity judged by their number of interactions. This is a non-personalized method to benchmark the recommendation performance [19].
- **BPR** [19]. This method uses the matrix factorization method with a pairwise ranking loss, which is tailored to learn to rank from implicit feedback. We report the best performance obtained by fixing and varying the learning rate.
- **eALS** [10]. This is a state-of-the-art matrix factorization method for item recommendation. It optimizes the squared loss (between actual item ratings and predicted ratings) and treats all unobserved interactions as negative instances and weighting them non-uniformly by item popularity.
- **NeuMF** [9]. This is a state-of-the-art neural matrix factorization model. It treats the problem of generating recommendations using implicit feedback as a binary classification problem. Consequently it uses the binary cross-entropy loss to optimize its model parameters.

Our method is based on user-item interactions, hence we mainly compare it with other user-item models. We leave out the comparison with other models like SLIM [17] and CDAE [28] because these are item-item models and hence performance difference may be caused by the user models for personalization.

4.3 Evaluation Protocol

To evaluate the performance of the recommended item we use the leave-one-out evaluation strategy which has been widely adopted in literature [1, 10, 19]. For each user we held-out her latest interaction as the test instance and utilized the remaining data for training. Since it is time consuming to rank all items for every user during evaluation, we followed the popular strategy [6, 14] that randomly samples 100 items that the user has not interacted with, ranking the test item among the 100 items. The performance of a ranked list is judged by Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) [8]. We truncated the rank list at 10 for both the metrics. As such, the HR@k intuitively measures whether the test item is present in the top-k list, and the NDCG accounts for the position of the hit by assigning higher scores to hits at top ranks. We calculated both metrics for each test user and reported the average score.

4.4 Parameter Learning

We use an Intel i7-6700 CPU @ 3.40GHz which has a RAM of 32GB and a Tesla K40c GPU. We implemented our proposed method using Keras [4]. We randomly divide the labeled set into training and validation set in a 4:1 ratio. We tuned the hyper-parameters of our model using the validation set. The proposed model and all its variants are learned by optimizing the log likelihood given by

[1] https://github.com/dhruvkhattar/RARE

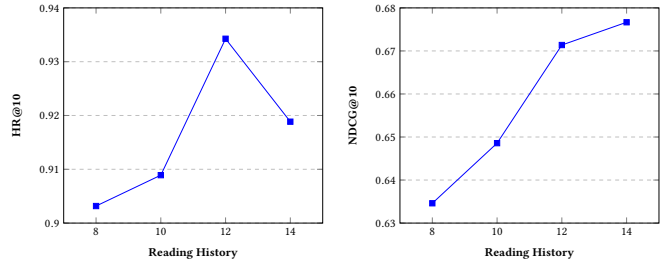

Figure 8: Performance of RARE w.r.t. the User's Reading History on NewsREEL

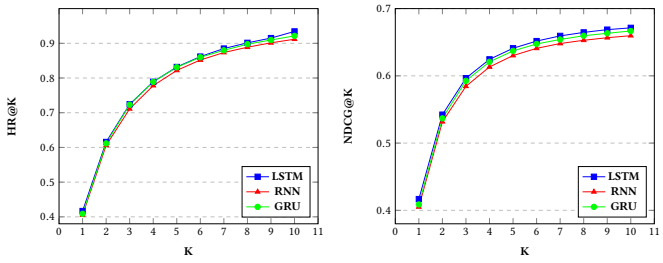

Figure 9: Performance of RARE w.r.t Recurrent Unit used in RARE on NewsREEL

Eq. 10. We initialize the fully connected network weights with the uniform distribution in the range between $-\sqrt{6/(fan_{in} + fan_{out})}$ and $\sqrt{6/(fan_{in} + fan_{out})}$ [7]. We used a batch size of 256 and used AdaDelta [29] as the optimizer.

5 Results and Analysis

In this section we present the results obtained by carrying different experiments with our method.

5.1 Performance Comparison with Baselines

For MF based methods like BPR and eALS, the number of predictive factors chosen is equal to the number of latent factors. We report the best performance in this case. For NeuMF, we vary the size of the CF layers (also latent factors) to choose the best fit for our model.

In Figs. 5 to 7, we compare our method with the baselines. Note that the performance of ItemPop measure was very weak and hence it does not show up clearly in the graphs. The Top-K recommended lists are used where K varies from 1 to 10. It is very clear from Figs. 5 and 6 that RARE outperforms other methods by a significant margin across all positions on the NewsREEL and the Malayalam datasets respectively. Although, RARE outperforms the other methods in case of Indonesian dataset as well (Fig. 7) but the margin is not that large. Amongst the different baselines, the trend in the performance can be seen as follows: NeuMF >eALS >BPR (in terms of both HR and NDCG). Although, in [19] it has been shown that BPR can be a strong performer for ranking performance owing to its pairwise ranking aware learner, we did not see the trend for our datasets. On the other hand RARE outperforms all the other baselines in terms of NDCG as well.

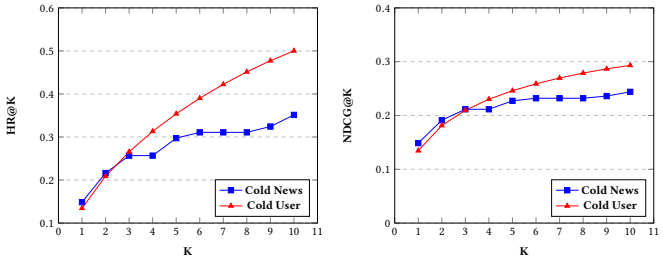

Figure 10: Performance of our model on Cold-Start cases

5.2 Effect of Size of Reading History

We vary the size of the reading history R used as inputs to our model. From Fig. 5, one can see that the Hit Ratio slowly increases with the size of the reading history until a certain point after which it decreases. However, the NDCG keeps on increasing. We can attribute this behaviour to the fact that users have diversified reading interests which only get effectively captured after a substantial number of interactions have been observed. However, after a while, increasing the user history often leads to over-specialization where the generic interests tend to overpower the specific ones. This is also an indicator of the fact that the preference of a user keeps varying and hence a window size should be chosen such that it helps the model to dynamically adapt to the users changing behaviour.

For all our methods, we chose a reading history of 12 for the users. We needed to make a choice between 12 and 14, and we chose 12 because we gave more importance to the HR rather than the NDCG.

5.3 Effect of different Encoders

We first note the effects on RARE by varying the kind of recurrent network used. We tested our model by using LSTMs, GRUs (Gated Recurrent Units) [5] and Vanilla RNN. From Fig. 9, the trend in the performance can be observed as follows: LSTM >GRU >RNN although the differences are not very large. One of the reasons for this could be the fact that an LSTM or a GRU is better able to encode the interests of the user as they handle long-term dependencies better.

We also note the effects when using different variants of our own model, i.e., when we replace the unified representation in RARE with solely the specific or the generic encoder. The results for this can be seen from Table 1. We note the trend in performance as follows RARE >Generic Encoder >Specific Encoder. This indicates that merely identifying the users' generic interests (a summary of overall interests) is not sufficient for learning a good recommendation model. However, when we use a combination of both in RARE, we find that the recommendation performance improves which clearly indicates that identifying both the specific and generic interests are essential for better recommendations.

5.4 Performance on Cold Start Cases

We then evaluated our model for the cold start cases as can be seen in Fig. 10. For this task we segregated users who had read a new news article in the end, i.e., they read articles which had never been seen before they read it. We found out that the number of such users were 74 in the CLEF dataset. There were very few such users

Table 1: Performance using different Encoding Mechanism on CLEF NewsREEL

Method	HR@10	NDCG@10
Specific Encoder	0.916	0.657
Generic Encoder	0.920	0.664
Specific + Generic (RARE)	**0.934**	**0.671**

Table 2: Performance of RARE by changing number of dense layers

Layers	HR@10	NDCG@10
128	0.913	0.659
128→64	**0.934**	**0.671**
128→64→32	0.912	0.666

in the other two datasets. Out of these 74 users, we see that the HR@10 is around 0.35. This promises us that our model is well suitable for handling the item cold-start problem.

For user cold-start, we test our learned model over users who had read articles in between 2 to 4 (inclusive) over the same dataset. Since we set the history size to 12, we had to set the remaining inputs to 0s. The HR@10 score was around 0.5. We see a gradual increase in the hit rates as we increase the value of K. The results promise the effectiveness of our model to handle the problem of user cold start as well. Although this is not exactly the user cold start problem because it still considers some number of user interactions, still it is worth noticing the performance because the baselines need a considerable amount of user history before making predictions. On the other hand, in our method, we can simply use the trained model for recommending articles to users who have had very few interactions.

5.5 Effect of Varying Layers

We observe the performance of our model when we vary the number of layers used in the Siamese Network in our model. We experiment by varying the number of layers along with the number of hidden units. We experiment by using one layer with size 128, two layers with sizes 128 and 64 and three layers with sizes 128, 64 and 32. From Table 2, we can see that the best performance is observed in the second case.

6 Conclusion

In this paper, we proposed the Recurrent Attentive News Recommendation Engine (RARE) to address the problem of news recommendation. We attempt to encode both the generic and the specific interests of the users. For the former we use a recurrent neural network while for the latter we use a recurrent network with an attention mechanism. We use the unified representations obtained from both these along with a Siamese network to make predictions. We conducted extensive experiments on three real-world datasets and demonstrated that our method can outperform the state-of-the-art methods in terms of different evaluation metrics.

References

[1] Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle. 2017. A Generic Coordinate Descent Framework for Learning from Implicit Feedback. In *WWW*.

[2] Robert M Bell and Yehuda Koren. 2007. Improved Neighborhood-based Collaborative Filtering. In *KDD*. 7–14.

[3] Minmin Chen, Zhixiang Xu, Fei Sha, and Kilian Q Weinberger. 2012. Marginalized Denoising Autoencoders for Domain Adaptation. In *ICML*. 767–774.

[4] François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv preprint arXiv:1412.3555* (2014).

[6] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In *WWW*. 278–288.

[7] Xavier Glorot and Yoshua Bengio. 2010. Understanding the Difficulty of Training Deep Feed-forward Neural Networks. In *AI-Stats*, Vol. 9. 249–256.

[8] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware Explainable Recommendation by Modeling Aspects. In *CIKM*. 1661–1670.

[9] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*.

[10] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 549–558.

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comp.* 9, 8 (1997), 1735–1780.

[12] Frank Hopfgartner, Torben Brodt, Jonas Seiler, Benjamin Kille, Andreas Lommatzsch, Martha Larson, Roberto Turrin, and András Serény. 2016. Benchmarking News Recommendations: The CLEF NewsREEL Use Case. In *ACM SIGIR Forum*, Vol. 49. 129–136.

[13] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. In *CIKM*. 2333–2338.

[14] Yehuda Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *KDD*. 426–434.

[15] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML*. 1188–1196.

[16] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 31–40.

[17] Xia Ning and George Karypis. 2011. Slim: Sparse Linear Methods for Top-n Recommender Systems. In *ICDM*. 497–506.

[18] Dinh Q Phung, Svetha Venkatesh, et al. 2009. Ordinal Boltzmann Machines for Collaborative Filtering. In *UAI*. 548–556.

[19] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[20] Jasson DM Rennie and Nathan Srebro. 2005. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*. ACM, 713–719.

[21] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *NIPS*, Vol. 1.

[22] Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th international conference on Machine learning*. ACM, 880–887.

[23] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*. ACM, 791–798.

[24] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 285–295.

[25] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 111–112.

[26] Florian Strub and Jeremie Mary. 2015. Collaborative Filtering with Stacked Denoising AutoEncoders and Sparse Inputs. In *NIPS Workshop on ML for eCommerce*.

[27] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NIPS*. 3104–3112.

[28] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-Encoders for Top-n Recommender Systems. In *WSDM*. 153–162.

[29] Matthew D Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701* (2012).

[30] Erheng Zhong, Nathan Liu, Yue Shi, and Suju Rajan. 2015. Building discriminative user profiles for large-scale content recommendation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2277–2286.