TrNews: Heterogeneous User-Interest Transfer Learning for News Recommendation

Guangneng Hu
HKUST / Hong Kong, China
njuhgn@gmail.com

Qiang Yang
HKUST / Hong Kong, China
qyang@cse.ust.hk

Abstract

We investigate how to solve the cross-corpus news recommendation for unseen users in the future. This is a problem where traditional content-based recommendation techniques often fail. Luckily, in real-world recommendation services, some publisher (e.g., Daily news) may have accumulated a large corpus with lots of consumers which can be used for a newly deployed publisher (e.g., Political news). To take advantage of the existing corpus, we propose a transfer learning model (dubbed as TrNews) for news recommendation to transfer the knowledge from a source corpus to a target corpus. To tackle the heterogeneity of different user interests and of different word distributions across corpora, we design a translator-based transfer-learning strategy to learn a representation mapping between source and target corpora. The learned translator can be used to generate representations for unseen users in the future. We show through experiments on real-world datasets that TrNews is better than various baselines in terms of four metrics. We also show that our translator is effective among existing transfer strategies.

1 Introduction

News recommendation is key to satisfying users' information need for online services. Some news articles, such as breaking news, are manually selected by publishers and displayed for all users. A huge number of news articles generated everyday make it impossible for editors and users to read through all of them, raising the issue of information overload. Online news platforms provide a service of personalized news recommendation by learning from the past reading history of users, e.g., Google (Das et al., 2007; Liu et al., 2010), Yahoo (Trevisiol et al., 2014; Okura et al., 2017), and Bing news (Lu et al., 2015; Wang et al., 2018).

When a new user uses the system (cold-start users) or a new article is just created (cold-start items), there are too few observations for them to train a reliable recommender system. Content-based techniques exploit the content information of news (e.g., words and tags) and hence new articles can be recommended to existing users (Pazzani and Billsus, 2007). Content-based recommendation, however, suffers from the issue of data sparsity since there is no reading history for them to be used to build a profile (Park and Chu, 2009).

Transfer learning is a common technique for alleviating the issues of data sparsity (Pan et al., 2010; Cantador et al., 2015; Liu et al., 2018). A user may have access to many websites such as Twitter.com and Youtube.com (Roy et al., 2012; Huang and Lin, 2016), and consume different categories of products such as movies and books (Li et al., 2009). In this case, transfer learning approaches can recommend articles to a new user in the target domain by exploiting knowledge from the relevant source domains for this new user.

A technical challenge for transfer learning approaches is that user interests are quite different across domains (corpora). For example, users do not use Twitter for the same purpose. A user may follow up on news about "Donald Trump" because she supports republican party (in the political news domain), while she may follow up account @*taylorswift13* ("Taylor Swift") because she loves music (in the entertainment news domain). Another challenge is that the word distribution and feature space are different across domains. For example, vocabularies are different for describing political news and entertainment news. An illustration is depicted in Figure 1. As a result, the user profile computed from her news history is heterogeneous across domains.

Several strategies have been proposed for heterogeneous transfer learning (Yang et al., 2009). The
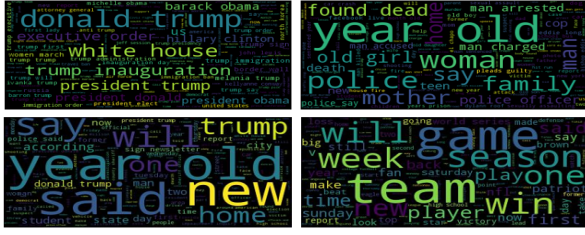
Figure 1: Word clouds of two news corpora. Top: Cheetah Mobile data. Bottom: MIND data. Left and right parts represent different domains (categories).

transferable contextual bandit (TCB) (Liu et al., 2018) learns a translation matrix to translate target feature examples to the source feature space. This linear mapping strategy is also used in collaborative cross networks (CoNet) (Hu et al., 2018) and deep dual transfer cross domain recommendation (DDTCDR) (Li and Tuzhilin, 2020). To capture complex relations between source and target domains, some nonlinear mapping strategy is considered in the embedding and mapping cross-domain recommendation (EMCDR) (Man et al., 2017) which learns a supervised regression between source and target factors using a multilayer perceptron (MLP). Since aligned examples between source and target domains are limited, they may face the overfitting issues.

To tackle challenges of heterogeneous user interests and limited aligned data between domains, we propose a novel transfer learning model (TrNews) for cross-corpora news recommendation. TrNews builds a bridge between two base networks (one for each corpus, see Section 3.1.1) through the proposed translator-based transfer strategy. The translator in TrNews captures the relations between source and target domains by learning a nonlinear mapping between them (Section 3.2). The heterogeneity is alleviated by translating user interests across corpora. TrNews uses the translator to transfer knowledge between source and target networks. TrNews alleviates the limited data in a way of alternating training (Section 3.3). The learned translator is used to infer the representations of unseen users in the future (Section 3.4). By "translating" the source representation of a user to the target domain, TrNews offers an easy solution to create unseen users' target representations. TrNews outperforms the state-of-the-art recommendation methods on four real-world datasets in terms of four metrics (Section 4.2), while having an explanation advantage by allowing the visualization of the importance

of each news article in the history to the future news (Section 4.4).

2 Related Work

Content recommendation Content-based recommendation exploits the content information about items (e.g., news title and article body (Yan et al., 2012; Xiao et al., 2019; Ma et al., 2019; Wu et al., 2020; Hu et al., 2020), tag, vlog (Gao et al., 2010)), builds a profile for each user, and then matches users to items (Lops et al., 2011; Yu et al., 2016; Wu et al., 2019b). It is effective for items with content or auxiliary information but suffers from the issues of data sparsity for users. DCT (Barjasteh et al., 2015) constructs a user-user similarity matrix from user demographic features including gender, age, occupation, and location (Park and Chu, 2009). NT-MF (Huang and Lin, 2016) constructs a user-user similarity matrix from Twitter texts. BrowseGraph (Trevisiol et al., 2014) addresses the freshly news recommendation by constructing a graph using URL links between web pages. NAC (Rafailidis and Crestani, 2019) transfers from multiple source domains through the attention mechanism. PdMS (Felício et al., 2017) assumes that there are many recommender models available to select items for a user, and introduces a multi-armed bandit for model selection. LLAE (Li et al., 2019a) needs a social network as side information for cold-start users. Different from the aforementioned works, we aim to recommending news to unseen users by transferring knowledge from a source domain to a target domain.

Transfer learning Transfer learning aims at improving the performance of a target domain by exploiting knowledge from source domains (Pan and Yang, 2009). A special setting is domain adaptation where a source domain provides labeled training examples while the target domain provides instances on which the model is meant to be deployed (Glorot et al., 2011; Li et al., 2019b). The coordinate system transfer (CST) (Pan et al., 2010) firstly learns the principle coordinate of users in the source domain, and then transfers it to the target domain in the way of warm-start initialization. This is equivalent to an identity mapping from users' source representations to their corresponding target representations. TCB (Liu et al., 2018) learns a linear mapping to translate target feature examples to the source feature space because there are many labelled data in the source domain. This linear
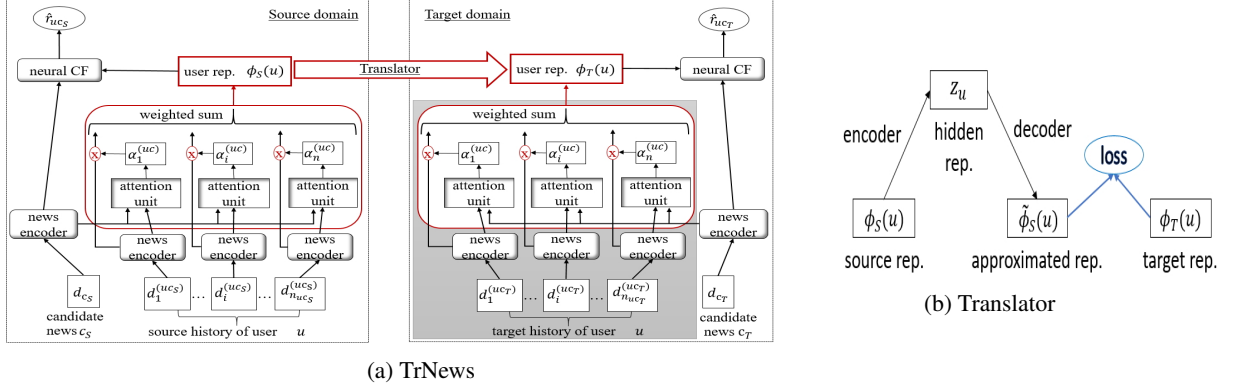
(a) TrNews

(b) Translator

Figure 2: Left: Architecture of TrNews. There is a base network for each of the two domains. The shaded area in the target network is empty for unseen users. The translator enables knowledge transfer between source and target networks. Right: The translator.

strategy is also used in CoNet (Hu et al., 2018) and DDTCDR (Li and Tuzhilin, 2020) which transforms the source representations to the target domain by a translation matrix. Nonlinear mapping strategy (Man et al., 2017; Zhu et al., 2018; Fu et al., 2019) is to learn a supervised mapping function between source and target latent factors by using neural networks. SSCDR (Kang et al., 2019) extends them to the semi-supervised mapping setting. Our translator is general to accommodate these identity, linear, and nonlinear transfer-learning strategies.

3 TrNews

3.1 Architecture

The architecture of TrNews is shown in Figure 2a, which has three parts. There are a source network for the source domain S and a target network for the target domain T, respectively. The source and target networks are both an instantiation of the base network (Section 3.1.1). The translator enables knowledge transfer between the two networks (Section 3.2). We give an overview of TrNews before introducing the base network and the translator.

Target network The information flow goes from the input, i.e., (user u, candidate news c_T) to the output, i.e., the preference score $\hat{r}_{uc_T}$, through the following three steps. First, the news encoder ψ_T computes the news representation from its content. The candidate news representation is $\psi_T(c_T) = \psi_T(d_{c_T})$ where d_{c_T} is c_T's content. The representations of historical news articles $[i]_{i=1}^{n_{uc_T}}$ of the user are $[\psi_T(d_i^{(uc_T)})]_{i=1}^{n_{uc_T}}$ where $d_i^{(uc_T)}$ is i's content and n_{uc_T} is size of the history. Second, the user encoder ϕ_T computes the user representation from her news history by: $\phi_T(u) = \phi_T\left([\psi_T(d_i^{(uc_T)})]_{i=1}^{n_{uc_T}}\right)$.

Third, the neural collaborative filtering (CF) module f_T computes the preference score by: $\hat{r}_{uc_T} = f_T([\phi_T(u), \psi_T(c_T)])$. We can denote the target network by a tuple (ψ_T, ϕ_T, f_T).

Source network Similarly to the three-step computing process in target network, we compute preference score $\hat{r}_{uc_S}$ from input (u, c_S) by: $\hat{r}_{uc_S} = f_S([\phi_S(u), \psi_S(c_S)])$ with tuple (ψ_S, ϕ_S, f_S).

Translator The translator $\mathcal{F}$ learns a mapping from the user's source representation to her target representation by $\mathcal{F}: \phi_S(u) \to \phi_T(u)$.

3.1.1 Base network

There is a base network for each of the two domains. It is an attentional network which has three modules (ψ, ϕ, f): the news encoder ψ to learn news representations, the user encoder ϕ to learn user representations, and a neural collaborative filtering module f to learn user preferences from reading behaviors.

News encoder The news encoder module is to learn news representation from its content. The news encoder takes a news article c's word sequence $d_c = [w_j]_{j=1}^{n_c}$ (n_c is length of c) as the input, and outputs its representation $\psi(c) \triangleq \psi(d_c) \in \mathbb{R}^D$ where D is the dimensionality. We compute the average of c's word embeddings by: $\psi(d_c) = \frac{1}{|d_c|} \sum_{w \in d_c} e_w$, where e_w is the embedding of w.

User encoder The user encoder module is to learn the user representation from their reading history. The user encoder takes a user's reading history $[\psi(d_i^{(u)})]_{i=1}^{n_u}$ (n_u is length of u's history) as input, and outputs her representation $\phi(u) \triangleq \phi([\psi(d_i^{(u)})]_{i=1}^{n_u}) \in \mathbb{R}^D$ where D is dimensionality.

In detail, given a pair of user and candidate news (u, c), we get the user representation $\phi(u|c)$ as the

weighted sum of her historical news articles' representations: $\phi(u|c) = \sum_{i=1}^{n_{uc}} \alpha_i^{(uc)} \psi(d_i^{(u)})$. The weights $\alpha_i^{(uc)}$'s are computed via attention units by: $\alpha_i^{(uc)} = a([\psi(d_i^{(u)}), \psi(d_c)])$ where a is the attention function with parameters to be learned. We use an MLP to compute it. For a specific candidate news c, we limit the history news to only those articles that are read before it. For notational simplicity, we do not explicitly specify the candidate news when referring to a user representation, i.e., $\phi(u)$ for short of $\phi(u|c)$.

Neural CF The neural collaborative filtering module is to learn preferences from user-news interactions. The module takes concatenated representations of user and news $[\phi(u), \psi(c)]$ as input, and outputs preference score $\hat{r}_{uc} = f([\phi(u), \psi(c)])$ where f is an MLP.

3.2 Translator

The target network suffers from the data sparsity issue of users who have no reading history. In this section, we propose a transfer learning component (i.e., the translator) to enable knowledge transfer for cross-domain news recommendation. The challenge is that user interests and word distributions are different across domains. For example, we compute the word clouds for two news corpora as shown in Figure 1. We can see that their word distributions are quite different and vocabularies are also different. Hence, user representations computed from their news history are heterogeneous across domains.

We build a translator, $\mathcal{F}: \phi_S(u) \rightarrow \phi_T(u)$, to learn a mapping from a user's source representation to her target representation as shown in Figure 2b. This translator captures the relationship and heterogeneity across domains. The translator learns to approximate the target representation from the source representation.

The translator takes a user's source representation $\phi_S(u)$ as the input, and maps it to a hidden representation z_u via an encoder parameterized by θ, and then gets a approximated representation $\tilde{\phi}_S(u)$ from it via a decoder parameterized by θ'. The parameters $\Theta_{\mathcal{F}} = \{\theta, \theta'\}$ of the translator are optimized to minimize the approximation error:

$$\mathcal{L}_{\mathcal{F}} = \frac{1}{|\mathcal{U}_0|} \sum_{u \in \mathcal{U}_0} ||H\tilde{\phi}_S(u) - \phi_T(u)||_2^2, \quad (1)$$

where $\mathcal{U}_0 = \mathcal{U}_S \cap \mathcal{U}_T$, and $\mathcal{U}_S$ and $\mathcal{U}_T$ are the user sets of source and target domains, respectively. H

is to match the dimensions of source and target representations.

Note that, we do not minimize the approximation error between $\phi_S(u)$ and $\tilde{\phi}_S(u)$ as with the standard autoencoder because our goal is to learn a mapping from a user's source representation to her corresponding target representation. After training, the learned mapping function is then used for inferring representations of unseen users in the target domain (the inference process will be described later in Section 3.4). It fulfills knowledge transfer from the source to the target domain via a supervised learning process.

Extensions The translator can be generalized to multiple, say k, source domains. We learn k translators using the aligned examples from each of the source domain to the target domain and then we average (or concatenate) the k mapped representations as the final representation for the user. Another extension is to introduce denoising or stacking techniques into the translator framework, not just the MLP structure in (Man et al., 2017).

3.3 Model learning

We learn TrNews in two stages. First, we train the source network using source training examples D_S and train the target network using target training examples D_T, respectively. Second, we train the translator by pairs of user representations computed on-the-fly from source and target networks. We introduce these two stages in detail.

First, TrNews optimizes the parameters associated with target network $\Theta_T = \{\theta_{\phi_T}, \theta_{\psi_T}, \theta_{f_T}\}$ and source network $\Theta_S = \{\theta_{\phi_S}, \theta_{\psi_S}, \theta_{f_S}\}$ by minimizing the joint cross-entropy loss:

$$\mathcal{L} = -\sum_{D_T}(r_{uc_T} \log \hat{r}_{uc_T} + (1 - r_{uc_T}) \log(1 - \hat{r}_{uc_T}))$$
$$-\sum_{D_S}(r_{uc_S} \log \hat{r}_{uc_S} + (1 - r_{uc_S}) \log(1 - \hat{r}_{uc_S})), \quad (2)$$

where the two terms on the right-hand side are to optimize losses over user-news examples in the target and source domains, respectively. They are related by the word embedding matrix for the union of words of the two domains. We generate D_T and D_S as follows and take the target domain as an example since the procedure is the same for the source domain. Suppose we have a whole news reading history for a user u, say $[d_1, d_2, ..., d_{n_u}]$. Then we generate the positive training examples by sliding over the history sequence: $D_T^+ = \{([d_i]_{i=1}^{c-1}, d_c) : c = 2, ..., n_u\}$. We adopt the random negative

sampling technique (Pan et al., 2008) to generate the corresponding negative training examples $D_T^- = \{([d_i]_{i=1}^{c-1}, d_c') : d_c' \notin [d_1, d_2, ..., d_{n_u}]\}$, that is, we randomly sample a news article from the corpus as a negative sample which is not in this user's reading history.

Second, TrNews optimizes the parameters associated with the translator $\Theta_{\mathcal{F}} = \{\theta, \theta'\}$ by Eq. (1). Since the aligned data is limited, we increase the training pairs by generating them on-the-fly during the training of the two networks, i.e., in an alternating way. The model learning is summarized in Algorithm 1.

Algorithm 1: Training of TrNews.

Input: $D_T, D_S, \mathcal{U}_0$
Output: Source & target networks, translator
for $iter = 1, 2, ..., 50$ **do**

    1. Train target and source networks with mini batch using D_T, D_S respectively;

    2. **for** $u \in \mathcal{U}_0$ **do**

        (a) Generate source representations $\phi(u_S)$ using source network;
        (b) Generate target representations $\phi(u_T)$ using target network;
        (c) Train the translator using pairs $(\phi(u_S), \phi(u_T))$ with mini batch;

    end

    3. Compute metrics on the validation set;

    if *No improvement for 10 iters* **then**
        Early stopping;
    end

end

3.4 Inference for unseen users

For a new user in the target domain (not seen in the training set $\mathcal{U}_T^{train}$), we do not have any previous history to rely on in learning a user representation for her. That is, the shaded area of the target network in Figure 2a is empty for unseen users.

TrNews estimates a new user u^*'s target representation by mapping from her source representation using the learned translator $\mathcal{F}$ by:

$$\phi_T(u^*) := \mathcal{F}(\phi_S(u^*)), \ \forall u^* \in \mathcal{U}_S \wedge u^* \notin \mathcal{U}_T^{train}, \quad (3)$$

where we compute $\phi_S(u^*)$ using u's latest reading history in the source domain. Then we can predict the user preference for candidate news c^* by:

$$\hat{r}_{u^*c^*} = f_T([\phi_T(u^*), \psi_T(c^*)]). \quad (4)$$

4 Experiment

We evaluate the performance of TrNews (Section 4.2) and the effectiveness of the translator (Section 4.3) in this section.

4.1 Datasets and experimental setup

Datasets We evaluate on two real-world datasets. The first $NY, FL, TX, \& CA$ are four sub-datasets extracted from a large dataset provided by an internet company Cheetah Mobile (Liu et al., 2018; Hu et al., 2019). The information contains news reading logs of users in a large geographical area collected in January of 2017, ranging from New York (NY), Florida (FL), Texas (TX), to California (CA) based on the division of user geolocation. They are treated as four rather than a single because the user set is not overlapped among them. The top two categories (political and daily) of news are used as the cross corpora. The mean length of news articles is around 12 words while the max length is around 50 words. The mean length of user history is around 45 articles while the max length is around 900 articles. The second $MIND$ is a benchmark dataset released by Microsoft for news recommendation (Wu et al., 2020). We use the MIND-small version to investigate the knowledge transfer when news reading examples are not so large and it is publicly available https://msnews.github.io/. The title and abstract of news are used as the content. The clicked historical news articles are the positive examples for user. The top two categories (news and sports) of news are used as the cross corpora. The word clouds of the two datasets are shown in Figure 1 and the statistics are summarized in Table 1. The mean length of news articles is around 40 words while the max length is around 123 words. Besides, the mean length of user history is around 13 articles while the max length is around 246 articles.

Evaluation protocol We randomly split the whole user set into two parts, training and test sets where the ratio is 9:1. Given a user in the test set, for each news in her history, we follow the strategy in (He et al., 2017) to randomly sample a number of negative news, say 99, which are not in her reading history and then evaluate how well the recommender can rank this positive news against these negative ones. For each user in the training set, we reserve her last reading news as the valid set. We follow the typical metrics to evaluate top-K news recommendation (Peng et al., 2016; Okura et al.,

Data	#user	Target domain			Source domain		
		#news	#reading	#word	#news	#reading	#word
NY	14,419	33,314	158,516	368,000	23,241	139,344	273,894
FL	15,925	33,801	178,307	376,695	25,644	168,081	340,797
TX	20,786	38,395	218,376	421,586	29,797	221,344	343,706
CA	26,981	44,143	281,035	481,959	32,857	258,890	375,612
MIND	25,580	9,372	211,304	461,984	8,577	120,409	346,988

Table 1: Statistics of the datasets.

NY	HR@5	HR@10	NDCG@5	NDCG@10	MRR	AUC
POP	52.96	67.66	40.34*	45.10	39.89*	77.92
LR	53.24	74.00	36.15	42.86	34.95	91.64
TANR	52.53	71.63	37.24	43.37	36.50	91.35
DeepFM	52.02	73.71	39.17	45.38	39.56	91.79
DIN	57.10*	75.66*	40.23	46.13*	38.65	92.29*
TrNews	**82.60**	**95.15**	**60.78**	**64.83**	**55.70**	**97.28**
FL	HR@5	HR@10	NDCG@5	NDCG@10	MRR	AUC
POP	52.45	66.14	39.72*	44.15	39.15*	79.33
LR	54.26*	73.90*	37.15	43.56	35.89	91.79
TANR	49.98	69.46	36.08	42.37	35.95	90.88
DeepFM	52.36	73.02	36.05	42.74	36.29	91.64
DIN	53.98	73.33	37.96	44.18*	36.96	91.86*
TrNews	**81.83**	**94.45**	**62.53**	**66.63**	**58.39**	**97.41**
TX	HR@5	HR@10	NDCG@5	NDCG@10	MRR	AUC
POP	54.21	67.87	40.62*	45.03*	39.64*	81.31
LR	55.72*	73.80*	39.24	44.97	37.78	91.74*
TANR	49.87	68.75	35.82	41.89	35.59	90.56
DeepFM	52.19	71.95	35.40	41.92	35.65	91.17
DIN	53.72	72.70	38.47	44.59	37.62	91.53
TrNews	**81.50**	**94.67**	**61.76**	**66.11**	**57.49**	**97.21**
CA	HR@5	HR@10	NDCG@5	NDCG@10	MRR	AUC
POP	58.32*	71.19	44.71*	48.86*	43.44*	83.38
LR	58.82	75.67*	42.16	47.65	40.44	92.37*
TANR	49.87	68.75	35.81	41.88	35.58	90.56
DeepFM	55.58	74.73	38.82	45.16	38.21	92.25
DIN	55.31	73.70	40.14	46.09	39.20	92.03
TrNews	**81.54**	**94.72**	**61.99**	**66.25**	**57.70**	**97.22**
MIND	HR@5	HR@10	NDCG@5	NDCG@10	MRR	AUC
POP	84.18	92.80	69.61	72.43	66.33	95.13
LR	92.69*	96.66*	85.81	87.11*	84.33*	97.92*
TANR	89.94	95.34	89.94*	83.38	79.84	97.86
DeepFM	89.16	94.78	79.36	81.19	77.12	97.63
DIN	89.28	94.88	80.16	82.03	78.22	97.63
TrNews	**97.36**	**99.02**	**94.16**	**94.74**	**93.45**	**99.47**

Table 2: Comparison of different recommenders.

2017; An et al., 2019) which are hit ratio (HR), normalized discounted cumulative gain (NDCG), mean reciprocal rank (MRR), and the area under the ROC curve (AUC). We report the results at cut-off $K \in \{5, 10\}$.

Implementation We use TensorFlow. The optimizer is Adam (Kingma and Ba, 2015) with learning rate 0.001. The size of mini batch is 256. The neural CF module has two hidden layers with size 80 and 40 respectively. The size of word embedding is 128. The translator has one hidden layer on the smaller datasets and two on the larger ones. The history is the latest 10 news articles.

4.2 Comparing different recommenders

In this section, we show the recommendation results by comparing TrNews with different state-of-the-art methods.

Baselines We compare with following recommen-

dation methods which are trained on the merged source and target datasets by aligning with shared users: **POP** (Park and Chu, 2009) recommends the most popular news. **LR** (McMahan et al., 2013) is widely used in ads and recommendation. The input is the concatenation of candidate news and user's representations. **DeepFM** (Guo et al., 2017) is a deep neural network for ads and recommendation based on the wide & deep structure. We use second-order feature interactions of reading history and candidate news, and the input of deep component is the same as LR. **DIN** (Zhou et al., 2018) is a deep interest network for ads and recommendation based on the attention mechanism. We use the news content for news representations. **TANR** (Wu et al., 2019a) is a state-of-the-art deep news recommendation model using an attention network to learn the user representation. We adopt the news encoder and negative sampling the same with TrNews.

Results We have observations from results of different recommendation methods as shown in Table 2. Firstly, considering that breaking and headline news articles are usually read by every user, the POP method gets competitive performance in terms of NDCG and MRR since it ranks the popular news higher than the other news. Secondly, the neural methods are generally better than the traditional, shallow LR method in terms of NDCG, MRR, and AUC on the four subdatasets. It may be that neural networks can learn nonlinear, complex relations between the user and the candidate news to capture user interests and news semantics. Considering that the neural representations of user and candidate news are fed as the input of LR, it gets competitive performance on MIND data. Finally, the proposed TrNews model achieves the best performance with a large margin improvement over all other baselines in terms of HR, NDCG, and MRR and also with an improvement in terms of AUC. It validates the necessity of accounting for the heterogeneity of user interests and word distributions across domains. This also shows that the base network is an effective architecture for news recommendation and the translator is effective to enable the knowledge transfer from the source domain to the target domain. In more detail, it is inferior by training a global model from the mixed source and target examples and then using this global model to predict user preferences on the target domain, as baselines do. Instead, it is good by training source

Approach	Transfer strategy	Formulation
CST (Pan et al., 2010)	Identity mapping	$\phi_T(u) = \phi_S(u)$
TCB (Liu et al., 2018)	Linear mapping	$\phi_T(u) = H\phi_S(u)$
DDTCDR (Li and Tuzhilin, 2020)		H is orthogonal
EMCDR (Man et al., 2017)	Nonlinear mapping	$\phi_T(u) = \text{MLP}(\phi_S(u))$

Table 3: Different transfer learning strategies.

NY	HR@5	HR@10	NDCG@5	NDCG@10	MRR	AUC
CST	81.04	94.37	59.04	63.56	54.19	96.94
TCB	82.18	94.92*	60.36*	64.46*	55.23*	97.28*
DDTCDR	82.27	94.90	59.82	63.90	54.51	97.25
EMCDR	82.44*	94.87	60.35	64.33	55.06	97.24
TrNews	**82.60**	**95.15**	**60.78**	**64.83**	**55.70**	**97.28**
FL	HR@5	HR@10	NDCG@5	NDCG@10	MRR	AUC
CST	79.29	93.91	59.03	63.60	54.74	97.07
TCB	81.51	**94.83**	62.06	66.33*	57.90*	97.40*
DDTCDR	81.39	94.63*	61.76	66.12	57.68	97.37
EMCDR	81.52*	94.47	62.14*	66.23	57.87	97.37
TrNews	**81.83**	94.45	**62.53**	**66.63**	**58.39**	**97.41**
TX	HR@5	HR@10	NDCG@5	NDCG@10	MRR	AUC
CST	78.74	94.20	58.53	63.48	54.56	96.92
TCB	80.68	94.12	61.06	65.38	56.97	97.10
DDTCDR	81.08	94.57	61.02	65.50	56.87	97.10
EMCDR	81.34*	**94.72**	**61.78**	66.11*	**57.59**	97.16*
TrNews	**81.50**	94.67*	61.76*	**66.11**	57.49*	**97.21**
CA	HR@5	HR@10	NDCG@5	NDCG@10	MRR	AUC
CST	79.92	93.71*	60.19	64.63	55.97	97.12
TCB	80.90*	93.71*	**62.32**	**66.45**	**58.35**	**97.36**
DDTCDR	80.22	93.47	61.42	65.72	57.44	97.25
EMCDR	80.53	93.33	62.04*	66.18	58.11*	97.30*
TrNews	**81.54**	**94.72**	61.99	66.25*	57.70	97.22
MIND	HR@5	HR@10	NDCG@5	NDCG@10	MRR	AUC
CST	96.93	98.49	93.83	94.34	93.09	99.32
TCB	97.41*	98.94	94.21*	94.72	93.43	99.41
DDTCDR	97.38	98.99	**94.25**	**94.78**	**93.49**	99.46*
EMCDR	**97.42**	99.01*	94.16	94.68	93.35	99.45
TrNews	97.36	**99.02**	94.16	94.74*	93.45*	**99.47**

Table 4: Comparison of different transfer strategies.

and target networks on the source and target domains, respectively, and then learning a mapping between them, as TrNews does.

4.3 Comparing different transfer strategies

In this section, we demonstrate the effectiveness of the translator-based transfer-learning strategy.

Baselines We replace the translator of TrNews with the transfer-learning strategies of baseline methods as summarized in Table 3. All baselines are state-of-the-art recommenders and capable of recommending news to cold-start users. Note that, the compared transfer-learning methods are upgraded from their original versions. We strengthen them by using the neural attention architecture as the base component. In their original versions, CST and TCB use matrix factorization (MF) while DDTCDR and EMCDR use multilayer perceptron. The neural attention architecture has shown superior performance over MF and MLP in the literature (Zhou et al., 2018; Wu et al., 2019a). As a result, we believe that the improvement will be larger if we compare with their original versions but this is obviously unfair.
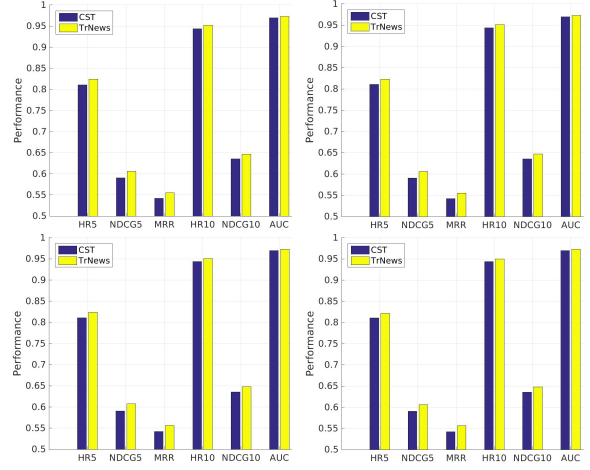

Figure 3: Impact of percentage (90%, 70%, 50%, 30%.) of shared users used to train the translator.

Results We have observations from results of different transfer learning strategies as shown in Table 4. Firstly, the transfer strategy of identity mapping (CST) is generally inferior to the linear (TCB and DDTCDR) and nonlinear (EMCDR and TrNews) strategies. CST directly transfers the source knowledge to the target domain without adaptation and hence suffers from the heterogeneity of user interests and word distributions across domains. Secondly, the nonlinear transfer strategy of EMCDR is inferior to the linear strategy of TCB in terms of MRR and AUC on the two smaller NY and FL datasets. This is probably because EMCDR increases the model complexity by introducing two large fully-connected layers in its MLP component. In contrast, our translator is based on the small-waist autoencoder-like architecture and hence can resist overfitting to some extent. Finally, our translator achieves the best performance in terms of NDCG, MRR and AUC on the two smaller NY and FL datasets, and achieves competitive performance on the two larger TX and CA datasets, and achieves the best performance in terms of HR and AUC on the MIND dataset, comparing with other four transfer methods. These results validate that our translator is a general and effective transfer-learning strategy to capture the diverse user interests accurately during the knowledge transfer for the unseen users in cross-domain news recommendation.

4.4 Analysis

Benefit of knowledge transfer We vary the percentage of shared users used to train the translator (see Eq. (1)) with {90%, 70%, 50%, 30%}. We compare with a naive transfer strategy of CST, i.e.,

Sharing?	HR@5	HR@10	NDCG@5	NDCG@10	MRR	AUC
No	81.31	94.69	59.43	63.72	54.37	97.16
Yes	**82.60**	**95.15**	**60.78**	**64.83**	**55.70**	**97.28**

Table 5: Impact of sharing word embeddings between source and target domains.

Strategy	HR@5	HR@10	NDCG@5	NDCG@10	MRR	AUC
Sepa.	82.36	94.86	60.62	64.65	55.51	97.28
Alter.	**82.60**	**95.15**	**60.78**	**64.83**	**55.70**	**97.28**

Table 6: Training TrNews with alternating (Alter.) vs separating (Sepa.) strategies.

(a) History length. (b) Embedding size.

Figure 4: Impact of the history length (left) and embedding size (right).

the way of direct transfer without adaptation. The results are shown in Figure 3 on the New York dataset. We can see that it is beneficial to learn an adaptive mapping during the knowledge transfer even when limited aligned examples are available to train the translator. TrNews improves relative 0.82%, 0.77%, 0.67%, 0.64% in terms of HR@10 performance over CST by varying among $\{90\%, 70\%, 50\%, 30\%\}$ respectively. So we think that the more aligned examples the translator has, the more benefits it achieves.

Impact of sharing word embeddings We investigate the benefits of sharing word embeddings between source and target domains. There is a word embedding matrix for each of the domains and we share the columns if the corresponding words occur in both domains. Take the New York dataset as an example, the size of the intersection of their word vocabularies is 11,291 while the union is 50,263. From the results in Table 5 we can see that it is beneficial to share the word embeddings even when only 22.5% words are intersected between them.

Impact of alternating training We adopt an alternating training strategy between training the two (source & target) networks and training the translator in our experiments. In this section, we compare this alternating strategy with the separating strategy which firstly trains the two networks and then trains the translator after completing the training of the two networks. That is, the training pairs of user representations for the translator are not generated on-the-fly during the training of source and target networks but generated only once after finishing their training. From the results in Table 6, we see that the alternating strategy works slightly better. This is probably because the aligned data between domains is limited and the alternating strategy increases the size of training pairs.

Impact of two-stage learning We adopt a two-stage model learning between training the two
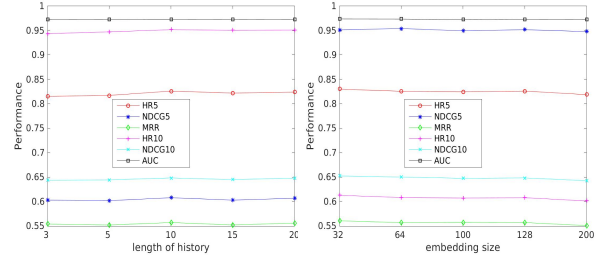
(source & target) networks and training the translator in our experiments. In this section, we compare this two-stage learning with an end-to-end learning which jointly trains the two networks and the translator. That is, the parameters of the translator depend on the word embedding matrix and on parameters of the user encoder. From the results in Table 7, we see that the two-stage learning works slightly better. This is probably because the aligned data between domains is too limited to reliably update the parameters which do not belong to the parameters of the translator.

Impact of the length of the history Since we generate the training examples by sliding over the whole reading history for each user, the length of reading history is a key parameter to influence the performance of TrNews. We investigate how the length of the history affects the performance by varying it with $\{3, 5, 10, 15, 20\}$. The results on the New York dataset are shown in Figure 4a. We can observe that increasing the size of the sliding window is sometimes harmful to the performance, and TrNews achieves good results for length 10. This is probably because of the characteristics of news freshness and of the dynamics of user interests. That is, the latest history matters more in general. Also, increasing the length of the input makes the training time increase rapidly, which are 58, 83, 143, 174, and 215 seconds when varying with $\{3, 5, 10, 15, 20\}$ respectively.

Impact of the embedding size In this section, we evaluate how different choices of some key hyperparameter affect the performance of TrNews. Except for the parameter being analyzed, all other parameters remain the same. Since we compute the news and user representations using the content of words, the size of word embedding is a key parameter to influence representations of words, uses, and news articles, and hence the perfor-

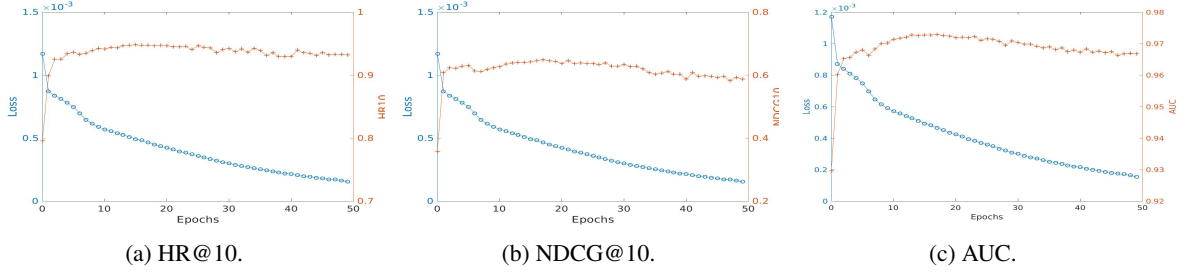|	(a) HR@10.	(b) NDCG@10.	(c) AUC.

Figure 5: Performance (right Y-axis in red cross) and loss (left Y-axis in blue circle) varying with training iterations.

mance of TrNews. We investigate how embedding size affects the performance by varying it with $\{32, 64, 100, 128, 200\}$. The results on the New York dataset are shown in Figure 4b. We can observe that increasing the embedding size is generally not harmful to the performance until 200, and TrNews achieves good results for embedding size 128. Changing it to 200 harms the performance a little bit since the model complexity also increases.

Optimization performance and loss We show the optimization performance and loss over iterations on the New York dataset in Figure 5. We can see that with more iterations, the training losses gradually decrease and the recommendation performance is improved accordingly. The most effective updates are occurred in the first 15 iterations, and performance gradually improves until 30 iterations. With more iterations, TrNews is relatively stable. For the training time, TrNews spends 143 seconds per iteration. As a reference, it is 134s for DIN and 139s for TCB, which indicates that the training cost of TrNews is efficient by comparing with baselines. Furthermore, the test time is 150s. The experimental environment is Tensorflow 1.5.0 with Python 3.6 conducted on Linux CentOS 7 where The GPU is Nvidia TITAN Xp based on CUDA V7.0.27.

Examining user profiles One advantage of TrNews is that it can explain which article in a user's history matters the most for a candidate article by using attention weights in the user encoder module. Table 8 shows an example of interactions between some user's history articles No. 0-9 and a candidate article No. 10, i.e., the user reads the candidate article after read these ten historical articles. We can see that the latest three articles matter the most since the user interests may remain the same during a short period. The oldest two articles, however, also have some impact on the candidate article, reflecting that the user interests may mix with a long-term characteristic. TrNews can capture these subtle short- and long-term user interests.

Training	HR@5	HR@10	NDCG@5	NDCG@10	MRR	AUC
end-to-end	81.85	94.63	60.58	64.74	55.68	97.03
two-stage	**82.60**	**95.15**	**60.78**	**64.83**	**55.70**	**97.28**

Table 7: Training TrNews in two-stage vs end-to-end.

No.	News title	Attn. weight
0	hillary clinton makes a low-key return to washington	0.04
1	the hidden message in obama's 'farewell' speech	0.12*
2	here's why sasha obama skipped the farewell address	0.00
3	donald trump's 'prostitute scandal' was filmed by cameras and recorded with microphones hidden behind the walls	0.00
4	white house official explains sasha obama's absence at father's farewell speech	0.00
5	irish bookie puts odds on trump's administration, inauguration and impeachment	0.00
6	heads are finally beginning to roll at the clinton foundation	0.00
7	donald trump's incoming administration considering white house without press corps	**0.76**
8	donald trump says merkel made 'big mistake' on migrants	0.05
9	controversial clinton global initiative closing its doors for good	0.00
10	army chief gen. bipin rawat talks about equal responsibility for women in the frontlines. we couldn't agree more	N/A

Table 8: Example I: Some articles matter more while some are negligible. (No. 10 is the candidate news)

5 Conclusion

We investigate the cross-domain news recommendation via transfer learning. The experiments on real-word datasets demonstrate the necessity of tackling heterogeneity of user interests and word distributions across domains. Our TrNews model and its translator component are effective to transfer knowledge from the source network to the target network. We also shows that it is beneficial to learn a mapping from the source domain to the target domain even when only a small amount of aligned examples are available. In future works, we will focus on preserving the privacy of the source domain when we transfer its knowledge.

Acknowledgement

References

Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long-and short-term user representations. In *Proceedings of Annual Meeting of Association for Computational Linguistics*, pages 336–345.

Iman Barjasteh, Rana Forsati, Farzan Masrour, Abdol-Hossein Esfahanian, and Hayder Radha. 2015. Cold-start item and user recommendation with decoupled completion and transduction. In *Proceedings of ACM Conference on Recommender Systems*.

I. Cantador, I. Fernández, S. Berkovsky, and P. Cremonesi. 2015. Cross-domain recommender systems. In *Recommender systems handbook*.

Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of World Wide Web*, pages 271–280.

Crícia Z Felício, Klérisson VR Paixão, Celia AZ Barcelos, and Philippe Preux. 2017. A multi-armed bandit model selection for cold-start user recommendation. In *Proceedings of Conference on User Modeling, Adaptation and Personalization*, pages 32–40.

Wenjing Fu, Zhaohui Peng, Senzhang Wang, Yang Xu, and Jin Li. 2019. Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 94–101.

Wen Gao, Yonghong Tian, Tiejun Huang, and Qiang Yang. 2010. Vlogging: A survey of videoblogging technology on the web. *ACM Computing Surveys*.

Xavier Glorot, Antoine Bordes, and Y. Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of international conference on machine learning*.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1725–1731.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of world wide web*.

Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of ACM International Conference on Information and Knowledge Management*, pages 667–676.

Guangneng Hu, Yu Zhang, and Qiang Yang. 2019. Transfer meets hybrid: a synthetic approach for cross-domain collaborative filtering with text. In *The World Wide Web Conference*, pages 2822–2829.

Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020. Graph neural news recommendation with unsupervised preference disentanglement. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4255–4264. Association for Computational Linguistics.

Yu-Yang Huang and Shou-De Lin. 2016. Transferring user interests across websites with unstructured text for cold-start recommendation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 805–814.

SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-supervised learning for cross-domain recommendation to cold-start users. In *Proceedings of ACM International Conference on Information and Knowledge Management*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *International Joint Conference on Artificial Intelligence*.

Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. 2019a. From zero-shot learning to cold-start recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4189–4196.

Pan Li and Alexander Tuzhilin. 2020. Ddtcdr: Deep dual transfer cross domain recommendation. In *Proceedings of ACM International Conference on Web Search and Data Mining*.

Yitong Li, T. Baldwin, and T. Cohn. 2019b. Semisupervised stochastic multi-domain learning using variational inference. In *Proceedings of Annual Meeting of Association for Computational Linguistics*.

Bo Liu, Ying Wei, Yu Zhang, Zhixian Yan, and Qiang Yang. 2018. Transferable contextual bandit for cross-domain recommendation. In *AAAI Conference on Artificial Intelligence*.

Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of international conference on Intelligent user interfaces*, pages 31–40.

Pasquale Lops, M. De Gemmis, and G. Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*.

Zhongqi Lu, Zhicheng Dou, Jianxun Lian, Xing Xie, and Qiang Yang. 2015. Content-based collaborative filtering for news topic recommendation. In *Twenty-ninth AAAI conference on artificial intelligence*.

Ye Ma, Lu Zong, Yikang Yang, and Jionglong Su. 2019. News2vec: News network embedding with subnode information. In *Proceedings of Conference on Empirical Methods in Natural Language Processing.*

Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-domain recommendation: an embedding and mapping approach. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 2464–2470.

Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. 2013. Ad click prediction: a view from the trenches. In *Proceedings of ACM SIGKDD international conference on Knowledge discovery and data mining.*

Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *IEEE International Conference on Data Mining*, pages 502–511.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer learning in collaborative filtering for sparsity reduction. In *AAAI conference on artificial intelligence.*

Seung-Taek Park and Wei Chu. 2009. Pairwise preference regression for cold-start recommendation. In *Proceedings of ACM conference on Recommender systems*, pages 21–28.

Michael Pazzani and D. Billsus. 2007. Content-based recommendation systems. In *The adaptive web.*

Hao Peng, Jing Liu, and Chin-Yew Lin. 2016. News citation recommendation with implicit and explicit semantics. In *Proceedings of Annual Meeting of Association for Computational Linguistics.*

Dimitrios Rafailidis and Fabio Crestani. 2019. Neural attentive cross-domain recommendation. In *Proceedings of ACM SIGIR International Conference on Theory of Information Retrieval*, pages 165–172.

Suman Deb Roy, Tao Mei, Wenjun Zeng, and Shipeng Li. 2012. Socialtransfer: cross-domain transfer learning from social streams for media applications. In *Proceedings of ACM international conference on Multimedia*, pages 649–658.

Michele Trevisiol, Luca Maria Aiello, Rossano Schifanella, and Alejandro Jaimes. 2014. Cold-start news recommendation with domain-dependent browse graph. In *Proceedings of ACM Conference on Recommender systems*, pages 81–88.

Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of World Wide Web Conference*, pages 1835–1844.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Yongfeng Huang, and Xing Xie. 2019a. Neural news recommendation with topic-aware news representation. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 1154–1159.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Tao Qi, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019b. Neural news recommendation with heterogeneous user behavior. In *Proceedings of Conference on Empirical Methods in Natural Language Processing.*

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606. Association for Computational Linguistics.

Wenyi Xiao, Huan Zhao, Haojie Pan, Yangqiu Song, Vincent W. Zheng, and Qiang Yang. 2019. Beyond personalization: Social content recommendation for creator equality and consumer satisfaction. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*

Rui Yan, Mirella Lapata, and Xiaoming Li. 2012. Tweet recommendation with graph co-ranking. In *Proceedings of Annual Meeting of Association for Computational Linguistics:*.

Qiang Yang, Yuqiang Chen, Gui-Rong Xue, Wenyuan Dai, and Yong Yu. 2009. Heterogeneous transfer learning for image clustering via the social web. In *Proceedings of Annual Meeting of Association for Computational Linguistics.*

Yang Yu, Xiaojun Wan, and Xinjie Zhou. 2016. User embedding for scholarly microblog recommendation. In *Proceedings of Annual Meeting of Association for Computational Linguistics.*

Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1059–1068.

Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, Mehmet Orgun, and Jia Wu. 2018. A deep framework for cross-domain and cross-system recommendations. In *International Joint Conference on Artificial Intelligence.*