MINER: Multi-Interest Matching Network for News Recommendation

Jian Li[1], Jieming Zhu[1], Qiwei Bi[2], Guohao Cai[1], Lifeng Shang[1], Zhenhua Dong[1], Xin Jiang[1], Qun Liu[1]
[1]Huawei Noah's Ark Lab
[2]School of Statistics, Renmin University of China
{lijian703, jamie.zhu, caiguohao1, shang.lifeng, dongzhenhua}@huawei.com
{jiang.xin, qun.liu}@huawei.com

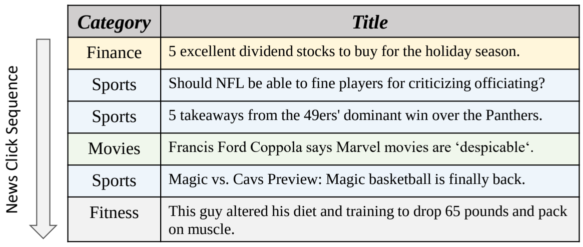
Abstract

Personalized news recommendation is an essential technique to help users find interested news. Accurately matching user's interests and candidate news is the key to news recommendation. Most existing methods learn a single user embedding from user's historical behaviors to represent the reading interest. However, user interest is usually diverse and may not be adequately modeled by a single user embedding. In this paper, we propose a *poly attention scheme* to learn multiple interest vectors for each user, which encodes the different aspects of user interest. We further propose a *disagreement regularization* to make the learned interests vectors more diverse. Moreover, we design a *category-aware attention weighting* strategy that incorporates the news category information as explicit interest signals into the attention mechanism. Extensive experiments on the MIND news recommendation benchmark demonstrate that our approach significantly outperforms existing state-of-the-art methods.

1 Introduction

Online news platforms such as Google News[1] and Microsoft News[2] have become a prevalent way for users to access news information (Das et al., 2007). The large amount of news generated every day make it hard for users to find their interested news. To alleviate information overload and improve reading experience, personalized news recommendation has become an essential part of these platforms (Liu et al., 2010; Phelan et al., 2011).

Accurate matching between users' interests and candidate news is the key to personalized news recommendation. Existing methods usually learn a *single* user interest vector by aggregating the previously browsed news via sequential or attentive models and then match it with candidate news vectors. For example, Okura et al. (2017) employ a GRU network to model user interest from the sequence of clicked news with the last hidden state of GRU being the user interest representation. An et al. (2019) also use a GRU network to aggregate clicked news sequence as an interest vector and combine it with the user ID embedding. Wu et al. (2019a) and Wu et al. (2019c) apply attentive pooling on the sequence of clicked news vectors to obtain user representations. However, user interest is usually varying and diverse. As the example shown in Figure 1, a user may be interested in different types of news (with distinct background colors) such as finance, sports, and movies. Therefore, it is insufficient for the above methods to accurately model user interest via a *single* user embedding, especially when the user has multiple interests with a long browsing history.


News Click Sequence

Category	*Title*
Finance	5 excellent dividend stocks to buy for the holiday season.
Sports	Should NFL be able to fine players for criticizing officiating?
Sports	5 takeaways from the 49ers' dominant win over the Panthers.
Movies	Francis Ford Coppola says Marvel movies are 'despicable'.
Sports	Magic vs. Cavs Preview: Magic basketball is finally back.
Fitness	This guy altered his diet and training to drop 65 pounds and pack on muscle.

Figure 1: The news click history of one user, who has various interests including finance, sports and movies.

In this paper, we propose a **M**ulti-**I**nterest Matching **N**etwork for n**E**ws **R**ecommendation (namely MINER), which can effectively capture the diverse nature of user's reading interests. Specifically, we first employ pre-trained BERT (Devlin et al., 2019) as the news encoder which is highly effective in modeling the text semantics. With the encoded news representation sequence, we propose a *poly attention scheme* to extract multiple interest vectors for each user. A matching score is calculated for each interest vector and the final matching score is aggregated by the individual scores. We study various aggregation methods including maximum, average, and weighted sum. Furthermore, to make the

[1] https://news.google.com/
[2] https://microsoftnews.msn.com/

Findings of the Association for Computational Linguistics: ACL 2022, pages 343 - 352
May 22-27, 2022 ©2022 Association for Computational Linguistics

extracted user interest representations more diverse, we propose a *disagreement regularization* (Li et al., 2021) which enlarges the distance among different interest vectors. In addition, news category information is usually available as shown in Figure 1, which reveals explicit user interest signals. To capture such signals, we propose a *category-aware attention weighting* strategy in the poly attention where historical news are re-weighted based on the category similarity to candidate news. We conduct extensive experiments and analysis on the real-world MIND news recommendation dataset (Wu et al., 2020), and the results show that MINER significantly outperforms the existing approaches.

The main contributions of this work can be summarized as follows:

- We propose a poly attention scheme in news recommendation to extract multiple interest vectors for each user. We further improve it with a disagreement regularization to make the extracted vectors more diverse.
- We propose a category-aware attention weighting strategy in the poly attention, which captures explicit category signals for user interest modeling.
- MINER achieves new state-of-the-art on the MIND benchmark and ranked the first on official leaderboard[3] in September 2021.

2 Related Work

2.1 Traditional Recommendation Methods

In recommender systems, most features are categorical and represented as IDs (e.g., itemID, cityID), leading to many studies that focus on modeling feature interactions. For example, FM (Rendle, 2012) models feature interactions with pairwise inner products. Wide&Deep (Cheng et al., 2016) and DeepFM (Guo et al., 2017) further make improvements by integrating both shallow and deep networks. DCN (Wang et al., 2017) models feature interactions via deep and cross sub-networks. Recent research pays more attention to the sequential recommendation problem, which aims to capture users' sequential behaviors via sequence modeling, such as RNN (Hidasi et al., 2016), CNN (Tang and Wang, 2018), and self-attention networks (Kang and McAuley, 2018; Sun et al., 2019). While most studies represent user via a single embedding vector, Li et al. (2019a) propose a capsule routing method (Sabour et al., 2017; Li et al., 2019b) to extract multiple user interest vectors. Yet, the model is specially designed for the matching stage of e-commerce recommendation. In contrast, we aim to learn users' multi-interest representations from news content via a novel poly attention scheme.

2.2 Neural News Recommendation

For news recommendation, traditional ID-based methods often suffer from the cold-start problem since news articles update very quickly (Wu et al., 2020). Consequently, many content-based methods explore neural networks to automatically learn and match news and user representations (Okura et al., 2017). For example, An et al. (2019) apply CNN to encode news and a GRU network to capture user interests from users' historical clicks. Attention mechanisms have been widely adopted in news recommendation to learn news and user representations, such as attentive multi-view learning (Wu et al., 2019a), personalized attention networks (Wu et al., 2019b), and multi-head self-attentions (Wu et al., 2019c). Some methods also incorporate knowledge graph information from news entities (Wang et al., 2018; Liu et al., 2020). Recent work have also applied the pre-trained BERT (Wu et al., 2021; Zhang et al., 2021) to encode news due to its superiority on text understanding. Yet, most methods learn a single user embedding which may not adequately model the diverse user interests. Accordingly, Qi et al. (2021) propose to utilize the news category labels to build hierarchical user interest representations. However, their representations are fixed at the three-level hierarchy. In contrast, the number of interest vectors in our MINER is a tunable hyper-parameter.

3 Our Approach

In this section, we first formulate the problem of personalized news recommendation. Then we introduce our proposed MINER in detail, whose overall framework is shown in Figure 2.

3.1 Problem Formulation

Given a user u and a candidate news n^c, our goal is to calculate the interest score s measuring the interest of user u in the content of news n^c. Then a set of candidate news N^c are ranked based on the interest scores and top ones are recommended to

[3] https://msnews.github.io/#leaderboard

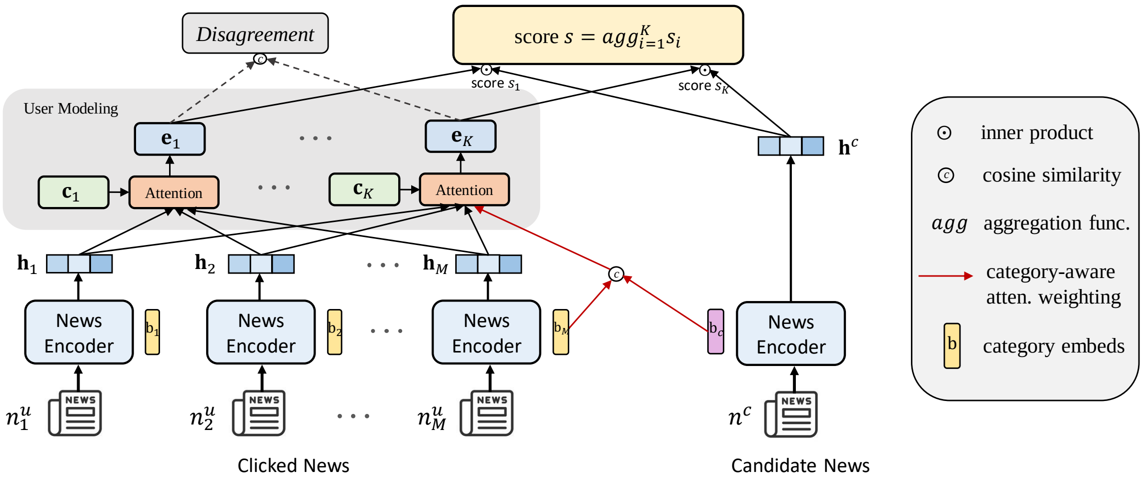


Figure 2: The overall framework of MINER, which consists of a news encoder, a multi-interest user modeling module, and a click score predictor. Disagreement regularization is introduced to make the multiple interest representations more diverse. Category-aware attention weighting is used to re-weight historical news according to the category similarity to candidate news.

user u. The user u consists of a list of historical clicked news $N^u = [n_1^u, n_2^u, ..., n_M^u]$, where M is the number of clicked news. Each news n is associated with its title texts T and a category ct.

3.2 News Encoder

News encoder is one of the core components in news recommendation that aims to learn the embeddings of news from their texts. It can be implemented by various NLP methods such as CNN (Kim, 2014) and Transformer (Vaswani et al., 2017). In this paper, we adopt the pre-trained BERT (Devlin et al., 2019) as news encoder, which can effectively capture the deep semantics of news texts. BERT has been successfully applied in various text ranking problems (Khattab and Zaharia, 2020; Karpukhin et al., 2020). Specifically, we feed tokenized news text into BERT model and use the output of `[CLS]` token as the news embedding $\mathbf{h}$. Thus the user u and candidate news n^c are encoded as $\mathbf{H}^u = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_M]$ and $\mathbf{h}^c$, respectively.

In ablation experiments (§4.4), we will also employ shallow word embeddings (Pennington et al., 2014) and self-attention networks to replace BERT.

3.3 Multi-Interest User Modeling

Another core component in news recommendation is user modeling, which receives a sequence of clicked news embeddings as input and outputs user representation $\mathbf{u}$ that summarizes user interest information. Traditionally, a *single* embedding vector is learned via sequential or attentive methods (An et al., 2019; Wu et al., 2019b). However, user interest is usually varying and diverse. We argue that representing user interests by one representation vector can be a bottleneck for news recommendation, since we have to compress all the information related with diverse interests of user into one representation vector. Instead, we propose to learn multiple representation vectors to express the distinct interests of user.

Specifically, we develop a *poly attention scheme* that extracts K interest vectors for each user through K additive attentions. Our method is inspired by the recently proposed Poly-Encoder (Humeau et al., 2020), and we generalize its idea from word sequence to user behavior sequence. In particular, we introduce K learnable context codes, i.e., $\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_K$, where each $\mathbf{c}_i$ extracts an interest embedding $\mathbf{e}_i$ by attending over the sequence of clicked news embeddings:

$$\mathbf{e}_i = \sum_{j=1}^{M} w_j^{c_i} \mathbf{h}_j, \quad w_j^{c_i} = \mathrm{softmax}(\phi_h^{c_i}(\mathbf{h}_j)), \quad (1)$$

where $w_j^{c_i}$ denotes the attention weight of the j-th historical news. $\phi_h^{c_i}(\cdot)$ is a dense network over the context code $\mathbf{c}_i$ and news representation $\mathbf{h}$:

$$\phi_h^{c_i}(\mathbf{h}_j) = \mathbf{c}_i^\top \tanh(\mathbf{W}^h \mathbf{h}_j), \quad (2)$$

where $\mathbf{c}_i$ and $\mathbf{W}^h$ are both trainable parameters.

In this way, we extract multiple user interest vectors $\mathbf{E}^u = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_K]$ with each representing certain aspect of user interests. Note the interest vectors are learned via soft attentions thus they may not have explicit meanings.

Disagreement Regularization Since the proposed *poly attention* aims to capture the distinct nature of user interests, it is beneficial to make the extracted interest representations more diverse. To this end, we further propose a *disagreement regularizaiton* (Li et al., 2018) to improve the poly attention, that enlarges the distance among different interest vectors during training. Specifically, we calculate the cosine similarity between each pair of interest vectors through the dot product of the normalized vectors. Then our training objective is to *minimize* the average cosine similarity (i.e., *maximize* the distance) among all interest vector pairs. The regularization term is formally expressed as:

$$\mathcal{L}_D = \frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{K}\frac{\mathbf{e}_i^\top \mathbf{e}_j}{\|\mathbf{e}_i\|\|\mathbf{e}_j\|}, \quad (3)$$

where K is the number of interest vectors.

Click Predictor For each interest vector $\mathbf{e}_i$, we calculate a matching score with the candidate news representation $\mathbf{h}^c$ via inner product:

$$s_i = \mathbf{e}_i^\top \mathbf{h}^c. \quad (4)$$

We propose several ways to aggregate the K matching scores as a final user click score, including:

- MINER-*max* takes the maximum value of the individual scores, i.e. $s = \max_{i=1}^{K} s_i$.
- MINER-*mean* takes the average value of the individual scores, i.e. $s = \text{mean}_{i=1}^{K} s_i$.
- MINER-*weighted* adopts a *target-aware attention* network (Wang et al., 2018) to weighted sum the individual scores according to the relevance between candidate news $\mathbf{h}^c$ and interest vector $\mathbf{e}_i$, i.e.:

$$s = \sum_{i=1}^{K} w_i s_i,$$
$$w_i = \text{softmax}(\mathbf{e}_i^\top \text{gelu}(\mathbf{W}^e \mathbf{h}^c)),$$

  where $\text{gelu}(\cdot)$ is the activation function and $\mathbf{W}^e$ is trainable parameter.

3.4 Category-aware Attention Weighting

In news recommendation dataset, category labels (e.g., Sports, Health) are usually available as shown in Figure 1. Besides the *implicit* user interests learned by soft attentions, the category information can be regarded as *explicit* user interest signals. Intuitively, a user tends to click certain categories of news. For example, the user in Figure 1 frequently clicks Sports news. Thus we can infer that he has a high probability to click another Sports news or similar type like Fitness news. Therefore, we propose a *category-aware attention weighting* strategy to re-weight historical news according to their category similarity to the candidate news, i.e., similar types of news have higher weights. [4]

Specifically, we first transfer the category words (e.g., Sports) of each news to word embedding through the pre-trained Glove (Pennington et al., 2014) vectors. Then we revise the attention weight $w_j^{c_i}$ over historical news in Equation 1 with an additional bias term:

$$w_j^{c_i} = \text{softmax}(\phi_h^{c_i}(\mathbf{h}_j) + \underline{\lambda \cos(\mathbf{b}_j, \mathbf{b}_c)}), \quad (5)$$

where $\mathbf{b}_j$ and $\mathbf{b}_c$ denote the category embedding of the j-th historical news and the candidate news. $\cos(\cdot)$ denotes the cosine similarity between the two category embeddings and λ is a learnable scalar. Note that, due to the exponential operation in *softmax* function, adding the original logit similarity $\phi_h^{c_i}(\mathbf{h}_j)$ with a bias term $\lambda \cos(\cdot)$ equals to multiplying the attention distribution by a scaling factor. In this way, we learn to re-weight the historical news according to category information.

3.5 Model Training

Following previous work (Huang et al., 2013; Wu et al., 2019c), we employ the NCE loss to train our ranking model. For each clicked news in the training dataset $\mathcal{D}$ which is termed as a positive sample n_i^+, we randomly select L non-clicked news in the same news session as negative samples $[n_i^1, ..., n_i^L]$. We then jointly predict the click scores of the positive news s^+ and L negative news $[s_i^1, ..., s_i^L]$. The loss $\mathcal{L}_{NCE}$ is the negative log-likelihood of all positive samples in $\mathcal{D}$:

$$\mathcal{L}_{NCE} = -\sum_{i=1}^{|\mathcal{D}|}\log\frac{\exp(s_i^+)}{\exp(s_i^+) + \sum_{j=1}^{L}\exp(s_i^j)}. \quad (6)$$

Together with the disagreement regularization in Equation 3, our final loss function is:

$$\mathcal{L} = \mathcal{L}_{NCE} + \beta * \mathcal{L}_D, \quad (7)$$

[4] We also tried simply concatenating the category embeddings with news embeddings, which underperforms the proposed method in this paper.

where β is a hyper-parameter and is set to 0.8 based on validation set performance.

4 Experiments

4.1 Experiment Setup

Dataset We evaluate our approach on a real-world news recommendation dataset MIND (Wu et al., 2020), which is collected from the user behavior logs of Microsoft News. There are two versions of the dataset, namely MIND-large and MIND-small. The MIND-large contains more than 15 million impression logs generated by 1 million users, from which the MIND-small randomly samples 50,000 users. An impression log records the clicked and non-clicked news that are displayed to a user at a specific time and his historical news click behaviors before this impression. Besides, MIND contains off-the-shelf category label of each news. Table 1 summarizes the data statistics.

	MIND-small	MIND-large
# News	65,238	161,013
# Categories	18	20
# Impressions	230,117	15,777,377
# Clicks	347,727	24,155,470

Table 1: Statistics of the two datasets.

Settings Following previous work (Wu et al., 2019b; Qi et al., 2021), we utilize users' most recent 50 clicked news to learn user representations. We only use news title for the experiments in this paper and the maximum length is set to 20.[5] The *bert-base-uncased* is used as the pre-trained model to initialize news encoders. The number of context codes K is set to 32 and we will show its influence in the analysis part. The dimension of the context code vectors is 200. The category embeddings are initialized by the 300-dimensional Glove (Pennington et al., 2014) vectors and are fixed during training. The negative sampling rate L is set to 4 during training, i.e., each positive news is paired with 4 negative news. We train MINER for 5 epochs with batch size being 128. The learning rate is set to $2e^{-5}$ and linearly decayed with 10% warmup steps. We employ Adam (Kingma and Ba, 2015) as the optimization algorithm. As previous work (Wu et al., 2020), we employ four ranking metrics, i.e., AUC, MRR, nDCG@5, and nDCG@10, for performance evaluation.

4.2 Comparison Baselines

We compare our proposed MINER against the following baseline methods:

Feature-based Methods: Traditional recommendation methods based on manual features and user-item interactions, including (1) *LibFM* (Rendle, 2012), that employs factorization machine on user ID, news ID, and TF-IDF features extracted from news titles; (2) *DeepFM* (Guo et al., 2017), a model combines factorization machine and deep neural network with the same features as *LibFM*.

Neural Recommendation Methods: Neural networks specially designed for news recommendation, including (1) *DKN* (Wang et al., 2018), using CNN to learn news representation and a target-aware attention network to learn user representation; (2) *NPA* (Wu et al., 2019b), using personalized attention networks on words and clicked news to learn news and user representations; (3) *NAML* (Wu et al., 2019a), using multi-view learning to obtain news representation and attentive pooling to learn user representation; (4) *LSTUR* (An et al., 2019), using a GRU network to learn short-term user interests and user ID embeddings as long-term interests; (5) *NRMS* (Wu et al., 2019c), leveraging multi-head self-attentions to learn news and user representations; (6) *HieRec* (Qi et al., 2021), learning hierarchical user interests including subtopic-level, topic-level, and user-level.

BERT-enhanced Methods: (1) Wu et al. (2021) apply BERT as the news encoder on several above methods. *LSTUR+BERT* and *NRMS+BERT* are included here; (2) *UNBERT* (Zhang et al., 2021), the SOTA news recommendation method with BERT that models multi-grained user-news matching.

We implement most baseline methods via the code and settings on Microsoft Recommenders[6].

4.3 Main Results

The overall performance of all baselines and three MINER variants (i.e. *-max, -mean, -weighted*) are summarized in Table 2. All the numbers in the table are percentage numbers with '%' omitted. The overall best and previously best results are boldfaced and underlined respectively. We have several observations from Table 2.

First, all neural news recommendations methods (Rows 3-14) consistently outperform manual feature-based methods (Rows 1-2). This is because

[5] For leaderboard submissions, we set the maximum title length as 32.

[6] https://github.com/microsoft/recommenders

		MIND-small				MIND-large			
#	Methods	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10
1	LibFM	59.74	26.33	27.95	34.29	61.85	29.45	31.45	37.13
2	DeepFM	59.89	26.21	27.74	34.06	61.87	29.30	31.35	37.05
3	DKN	62.90	28.37	30.99	37.41	64.07	30.42	32.92	38.66
4	NPA	64.65	30.01	33.14	39.47	65.92	32.07	34.72	40.37
5	NAML	66.12	31.53	34.88	41.09	66.46	32.75	35.66	41.40
6	LSTUR	65.87	30.78	33.95	40.15	67.08	32.36	35.15	40.93
7	NRMS	65.63	30.96	34.13	40.52	67.66	33.25	36.28	41.98
8	HieRec†	67.95	32.87	36.36	42.53	69.03	33.89	37.08	43.01
9	LSTUR+BERT‡	68.28	32.58	35.99	42.32	69.49	34.72	37.97	43.70
10	NRMS+BERT‡	68.60	32.97	36.55	42.78	69.50	34.75	37.99	43.72
11	UNBERT§	67.62	31.72	34.75	41.02	70.68	35.68	39.13	44.78
12	MINER-*max*	67.39	32.37	35.93	42.11	69.97	35.03	38.37	44.05
13	MINER-*mean*	69.49	33.44	37.37	43.53	71.37	36.06	39.56	45.21
14	MINER-*weighted*	**69.61**	**33.97**	**37.62**	**43.90**	**71.51**	**36.18**	**39.72**	**45.34**

Table 2: Performance of different methods. Previously best results are underlined (the higher, the better) and MINER significantly outperforms all baselines ($p < 0.01$). †: results are from Qi et al. (2021). ‡: results on MIND-large are from Wu et al. (2021). §: results are from Zhang et al. (2021). Our ensemble model on MIND-large ranked the first on official leaderboard: `https://msnews.github.io/#leaderboard` in September 2021.

the handcrafted features may not be optimal and the neural networks can learn implicit semantic features to better model the news and users.

Second, BERT-enhanced methods (Rows 9-11) perform generally better than traditional neural methods that are based on word embeddings (Rows 3-8). The reason is that the deeply stacked and large-scale pre-trained BERT model can better model text semantics than the shallow word embeddings, which is crucial for contents understanding in news recommendation. For example, *LSTUR+BERT* and *NRMS+BERT* significantly outperform *LSTUR* and *NRMS*, respectively.

Third, among the three MINER variants (Rows 12-14), MINER-*weighted* performs the best. This is because MINER-*weighted* incorporates the candidate news signal to adaptively select important interest vectors. Note MINER-*mean* slightly underperforms MINER-*weighted* but outperforms MINER-*max*. Potential reason is that one candidate news may match multiple extracted interests (e.g., diet news matches Health and Food), and the overall assessment based on all the interest vectors would be more accurate than matching a single one.

Last, MINER significantly outperforms other baseline methods in terms of all metrics on the two datasets. The significant improvements can be attributed to the multi-interests modeling and BERT news encoder. Other BERT-enhanced methods such as *LSTUR+BERT* and *NRMS+BERT* only lean a single user embedding to represent user interests, whose expressiveness may be insufficient. Instead, MINER learns multiple representation vectors to express the diverse user interests. Compared against *UNBERT* that concatenates all the history news and candidate news as BERT input, MINER is more flexible as *UNBERT* is restricted by the maximum length of BERT input. Note *HieRec* also incorporates category labels to build hierarchical user interests but it is fixed at the three-level interests hierarchy. In contrast, the number of interest vectors in MINER (i.e. K) is a tunable hyper-parameter.

Model	AUC	MRR	nDCG@10
HieRec (Qi et al., 2021)	67.95	32.87	42.53
MINER w/o BERT	68.07	32.93	42.62
w/o disagreement	67.42	32.38	42.12
w/o category	67.13	32.06	41.73
MINER with BERT	69.61	33.97	43.90
w/o disagreement	69.49	33.46	43.56
w/o category	69.38	33.60	43.60

Table 3: Effects of different MINER components.

4.4 Ablation Study

In this section, we study the effectiveness of different MINER components by removing them. The results on MIND-small are illustrated in Table 3.

We first show the effect of deeply stacked BERT encoder (§3.2) by replacing it with shallow word embeddings (Pennington et al., 2014). For a fair comparison, we take the SOTA non-BERT model *HieRec* (Qi et al., 2021) as reference and implement their knowledge-aware news encoder in our

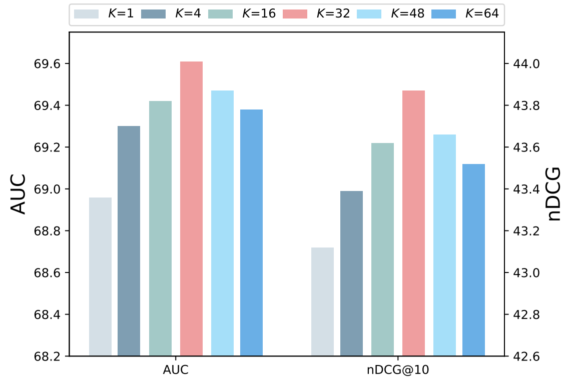


Figure 3: Influence of the number of interest vectors.

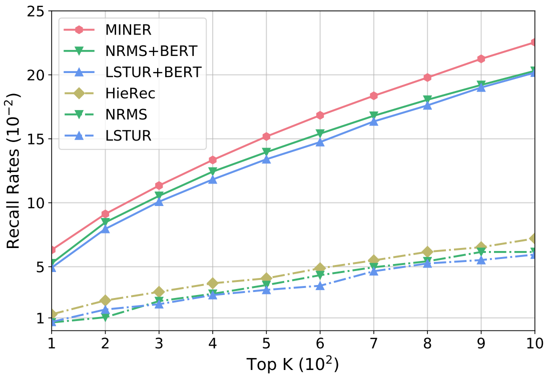


Figure 4: The performance on news recall.

MINER framework. The encoder consists of an word embedding layer, an entity embedding layer, and self-attention networks to learn text representation. The results in Table 3 demonstrate that BERT plays a crucial role in MINER and the performance largely decreases if we employ word embeddings. Nevertheless, this variant model is still able to outperform the SOTA non-BERT model *HieRec*, demonstrating the superiority of multi-interest user modeling and other components.

We further remove the proposed disagreement regularization (§3.3) and category-aware attention weighting (§3.4) from our MINER framework, respectively. The decreased performances in Table 3 respectively verify the benefits of diversifying the extracted interest vectors and incorporating category as explicit interest signals. Note MINER w/o BERT (above the dashed line) suffers more performance drop than MINER with BERT (below the dashed line). Potential reason is that the two techniques may have some overlapping effects with BERT thus it is hard to get further performance gain when BERT has already improved a lot. Besides, the performance drop from removing category-aware attention weighting is much larger than removing disagreement regularization, demonstrating the importance of category signals.

4.5 Number of Interest Vectors

In this section, we show the influence of hyper-parameter K, i.e., the number of extracted interest vectors in MINER. We vary this number and plot the results on MIND-small in Figure 3. We can see that with the increase of value K, the news recommendation performances first go up and then decline. The best results are achieved when $K = 32$. We conjecture the reason is that the expressiveness of MINER gradually increases when K is increasing. However, too large K may introduce more redundant parameters thus harm the overall performance. Besides, the best value of K may depend on the length of news browsing history M, and the longer browsing history requires more model capacity thus larger value of K. Note that when we set $K = 1$, i.e., MINER degrades to single user embedding, we still obtain a good AUC of 68.92 and we attribute it to the use of BERT news encoder and category-aware attention weighting.

4.6 Effectiveness on News Recall

To further verify the effectiveness of MINER , we also conduct experiments on news recall (or news retrieval). Instead of ranking a given list of candidate news, the aim of news recall is to retrieve certain number of candidate news from a large news pool which is usually the first stage in news recommendation. Therefore, efficiency is a key issue in news recall task. Conventional way is to build a bi-encoder model to *decouple* the modeling of user interests and candidate news thus all the news representations can be pre-computed and cached. Accordingly, we employ MINER-*mean* without the category-aware attention weighting and average the extracted interest vectors for news recall.

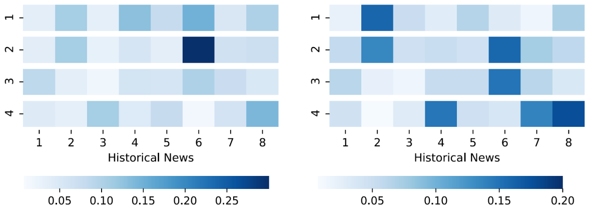
Following Qi et al. (2021), we do experiment on MIND-small and report the accuracy of top K recalled candidate news (i.e., recall rate) of each method. The results are shown in Figure 4 where the conclusions are generally consistent with Table 2. First, incorporating pre-trained BERT as news encoder significantly improves the recall rates, due to its superiority on text semantic modeling. Second, our MINER consistently achieves the best performance compared to other baseline methods. This is because MINER extracts user interests from multiple aspects, which is more ex-

		Historical Clicked News
1	Finance	Man who inherited 6 figures shares advice he'd give his younger self.
2	Sports	Foles will start for Jaguars over Minshew after bye week.
3	Sports	Pete Carroll takes swipe at Patriots over their strict culture.
4	Food	The best Trader Joe's desserts of all time.
5	Politics	Senate to try to override Trump emergency declaration veto Thursday.
6	Sports	NFL had no choice but to send a clear message with Garrett punishment.
7	Sports	Umpire Jeff Nelson leaves game with concussion after being hit by foul balls.
8	Food	Wendy's is turning 50 years old, and is gifting us free food through 2020.

	Recommended by NRMS+BERT
Sports	NFL week 8 power rankings: old-school football rules the day.
Sports	Patriots wanted a test. Now, they need some answers.
Politics	40 conservative groups sign ethics complaint against Pelosi.

	Recommended by MINER
Sports	Patriots wanted a test. Now, they need some answers.
Food	**National Dessert Day: Where to get free dessert at Wendy's.**
Health	Simple diet changes helped this guy lose 75 pounds in 9 months.

Figure 5: Case study on top 3 news recommended by *NRMS+BERT* and MINER in a sampled impression. The news actually clicked by the user is highlighted in blue.



(a) w/o disagreement

(b) with disagreement

Figure 6: Visualize the attention weights on the historical news in Figure 5 (a) before and (b) after applying disagreement regularization.

pressive than methods like *NRMS* and *LSTUR* that only learn a single user interest representation.

4.7 Case Study and Visualization

We conduct a case study to further shed light on the effectiveness of MINER. We compare MINER with *NRMS+BERT* since it is effective and also employs BERT news encoder but with single user embedding. In Figure 5, we show a sampled impression where the user has previously clicked 8 news. The news category is listed before the news title. We also show the top 3 recommended news by the two methods and the actually clicked news. We can see that both *NRMS+BERT* and MINER rank Sports news on the top, since the user has frequently clicked Sports news in the history. However, *NRMS+BERT* only learns a single user interest representation thus it is hard to capture other user interests. In contrast, our MINER extracts user interests from multiple aspects through the poly attention scheme, which can effectively find that the user is also keen to Food news. So MINER ranks a Food news in the second which is actually clicked by the user. Besides, MINER also recommends a Health news in the third place which is a related category to the Sports and Food, and we attribute this to our proposed category-aware attention weighting.

In addition, to show the effectiveness of disagreement regularization, we plot the attention map of this case in Figure 6. Specifically, we train a MINER with 4 interest vectors (i.e., $K = 4$) and visualize the attention weights (as Equation 5) before and after applying disagreement regularization. The vertical axis represents each interest extractor (i.e., additive attention) and the horizontal axis denotes the attention weights on historical news. We can find that before applying disagreement regularization, the attentions are mostly focused on the second and the sixth news which are Sports news, and the four attention distributions are quite similar. However, after the employment of disagreement, the four attention distributions become more *discriminative* and *diverse*, explicitly focusing on more news such as the fourth and the eighth news that are in the Food category.

5 Conclusion

In this paper, we propose a news recommendation method named MINER to capture the diver user interests from the historical reading behaviors, rather than most existing methods that learn a single user embedding to represent the reading interest. Specifically, we propose a *poly attention scheme* to learn multiple user interest vectors through soft attentions, which encode the different aspects of user interest. We further propose a *disagreement regularization* to improve the poly attention, that makes the learned interests vectors more diverse. Moreover, we design a *category-aware attention weighting* strategy to re-weight historical news according to the category similarity. Extensive experiments on the MIND news recommendation benchmark demonstrate the superiority of MINER over existing state-of-the-art methods. In addition, MINER ranked the first on the MIND leaderboard in September 2021.

Future work includes extending MINER to multimodal and multi-task scenarios (Bi et al., 2022).

6 Acknowledgements

We thank the anonymous reviewers for their insightful comments. We also appreciate the helpful discussion with the colleagues in our team.

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 *ACL*, pages 336–345.

Qiwei Bi, Jian Li, Lifeng Shang, Xin Jiang, Qun Liu, and Yang Hanfang. 2022. Mtrec: Multi-task learning over bert for news recommendation. In *Findings of ACL*.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & deep learning for recommender systems. In *(DLRS@RecSys*, pages 7–10.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. Deepfm: a factorization-machine based neural network for ctr prediction. In *IJCAI*.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *ICLR*.

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*, pages 2333–2338.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *ICLR*.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*, pages 197–206.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *SIGIR*, pages 39–48.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019a. Multi-interest network with dynamic routing for recommendation at tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2615–2623.

Jian Li, Zhaopeng Tu, Baosong Yang, Michael R Lyu, and Tong Zhang. 2018. Multi-head attention with disagreement regularization. In *EMNLP*, pages 2897–2903.

Jian Li, Xing Wang, Zhaopeng Tu, and Michael R Lyu. 2021. On the diversity of multi-head attention. *Neurocomputing*, 454:14–24.

Jian Li, Baosong Yang, Zi-Yi Dou, Xing Wang, Michael R Lyu, and Zhaopeng Tu. 2019b. Information aggregation for multi-head attention with routing-by-agreement. In *NAACL*, pages 3566–3575.

Danyang Liu, Jianxun Lian, Shiyin Wang, Ying Qiao, Jiun-Hung Chen, Guangzhong Sun, and Xing Xie. 2020. Kred: Knowledge-aware document representation for news recommendations. In *RecSys.*, pages 200–209.

Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *IUI*, pages 31–40.

Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *KDD*, pages 1933–1942.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Owen Phelan, Kevin McCarthy, Mike Bennett, and Barry Smyth. 2011. Terms of a feather: Content-based news recommendation and discovery using twitter. In *ECIR*, pages 448–459.

Tao Qi, Fangzhao Wu, Chuhan Wu, Peiru Yang, Yang Yu, Xing Xie, and Yongfeng Huang. 2021. Hierec: Hierarchical user interest modeling for personalized news recommendation. In *ACL*, pages 5446–5456.

Steffen Rendle. 2012. Factorization machines with libfm. *ACM TIST*, 3(3):1–22.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *NeurIPS*.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*, pages 1441–1450.

Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*, pages 565–573.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 6000–6010.

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

Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *ADKDD*, pages 12:1–12:7.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019a. Neural news recommendation with attentive multi-view learning. In *IJCAI*, pages 3863–3869.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019b. Npa: Neural news recommendation with personalized attention. In *KDD*, pages 2576–2584.

Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019c. Neural news recommendation with multi-head self-attention. In *EMNLP*, pages 6390–6395.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering news recommendation with pre-trained language models. In *SIGIR*, page 1652–1656.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020. Mind: A large-scale dataset for news recommendation. In *ACL*, pages 3597–3606.

Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, Jieming Zhu, Zhaowei Wang, and Xiuqiang He. 2021. Unbert: User-news matching bert for news recommendation. In *IJCAI*, pages 3356–3362.