Accuracy meets Diversity in a News Recommender System

Shaina Raza[1*], Syed Raza Bashir[2], Usman Naseem[3]
[1] University of Toronto, Toronto, ON, Canada
[2] Toronto Metropolitan University, Toronto, ON, Canada
[3] The University of Sydney, Sydney, Australia
shaina.raza@utoronto.ca, razabashir55@hotmail.com, usman.naseem@sydney.edu.au

Abstract

News recommender systems face certain challenges. These challenges arise due to evolving users' preferences over dynamically created news articles. Diversity is necessary for a news recommender system to expose users to a variety of information. We propose a deep neural network based on a two-tower architecture that learns news representation through a news item tower and users' representations through a query tower. We introduce diversity in the proposed architecture by considering a category loss function that aligns items' representation of uneven news categories. Experimental results on two news datasets reveal that our proposed architecture is more effective compared to the state-of-the-art methods and achieves a balance between accuracy and diversity.

1 Introduction

Reading the news has never been more common in people's daily lives than it is now. Big names like Yahoo!, Google, and CNN have launched online news portals that users can access from anywhere to browse various news categories and find up-to-date information. However, finding the right content is a challenge. With so much information available online, selecting relevant news has become a time-consuming and challenging task. A news recommender system (NRS) offers solutions to the information overload problem and provides relevant and interesting news recommendations to users (Raza and Ding, 2021b).

In the state-of-the-art of NRS, the news stories that users have read in the past are used to infer their interests and preferences. However, there is usually frequent content updating in a news domain. We show the news reading behaviour of a typical user in Figure 1 as an example.

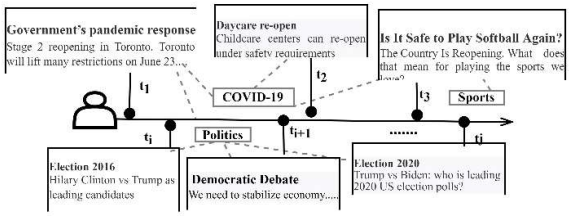

Figure 1: A user's behaviour during news reading

We see in Figure 1 that this user reads about the *'COVID-19 cases in Ontario'* during time t_1, and then read about *'Canadian children 5 to 11 could become eligible for COVID-19 vaccine…*" during t_2, and then his following action (read/click) is reading the news '*COVID-19 travel ban on tourists*' during t_3. We also observe that there is a transition in user preferences to other topics, for example, the elections, and politics in the later time. This shows that a user in an NRS generally changes his/her interests over time. Some of the user interests are long-term which shows a habit or personality, while some of the user interests are short-term, which may erupt due to some trending news, a sudden interest in an event and so. It is important to consider the accuracy as well as the diversity of news items while providing recommendations to users in an NRS (Raza and Ding, 2021a). This is a motivation for this research.

A common practice in the recommender systems is to design a two-tower architecture (Wang et al., 2019), where, first, a retrieval model retrieves a subset of related items from a large corpus in response to a user's query, and, then a ranking model ranks the retrieved items based on users' actions (clicks or ratings). The quality of retrieved items plays a critical role in the retrieval stage. The retrieval and ranking mechanism is also used in two-tower architectures (Yang et al., 2020).

In a typical two-tower architecture (Yi et al., 2019; Yang et al., 2020; Wang et al., 2019), there is usually no mechanism for the information interaction between the two towers. Usually, the

Proceedings of the 29th International Conference on Computational Linguistics, pages 3778–3787
October 12–17, 2022.

items or users are represented by their IDs or titles (for example, movie ID, news ID, user ID) to represent each tower. By missing the chance to include rich information related to news items (e.g., news body, categories, etc.) or user contexts (e.g., a situation when the user interacts with the news, such as time, and place) in the two-tower architecture, we are providing recommendations that are not quite relevant to users' preferences.

An issue with a typical recommender system is that they only focus on the relevancy of users' preferences and there is usually no consideration of diversity aspects while making recommendations (Raza and Ding, 2021a). For example, if we consider a real-time news recommendation scenario, we find that there are different categories of news (for example, sports, entertainment, politics, weather, and such), and the number of news items in each category varies (i.e., are imbalanced). As a result, the items in a single news category may account for most news recommendations. Consequently, the news recommendations that are produced may be too narrow and users' diversified interests are not addressed.

To address the above issues (incorporating rich user-item interactions and diversity of recommendations), we present a novel two-tower architecture for an NRS. We summarize our contributions as:

- We present a two-tower architecture for NRS and supplement the item tower with rich side information (meta-data) related to news items. We also consider the user context(s) in the query tower. We represent each user query by an augmented vector, which consists of the user's query and the news item features. The augmented vector is then updated based on the output representation vector of the other tower for a positive sample. In this way, the augmented vector implicitly models the information interaction between the two towers.

- To introduce diversity in the two-tower architecture, we include a category loss function during the training phase that aligns the representation of news items from a variety of news categories.

Extensive experiments on two news datasets show that our proposed approach has two major advantages: (i) it provides deeper insights into the information interaction of two-tower models in an NRS, and (ii) it provides diversified news recommendations (along with relevant recommendations) from a variety of news categories. Our goal is to provide news recommendations that are relevant to users' past preferences (interests) and are diversified at the same time

2 Related Work

Deep learning has demonstrated great success in recommender systems, such as in movie recommendations (WeAreNetflix, 2018), social networks (Ojagh et al., 2020) and many other application domains. Recommending news is particularly challenging (Raza and Ding, 2021b). This is because of the dynamic nature of the news domain and changing users' preferences. The state-of-the-art NRSs (Wang et al., 2018; An et al., 2019; Zhu et al., 2019) has shown tremendous performance in both academia and industry, however, a few challenges need to be addressed. First, these models do not extract enough news data from a reader's history. There are many pieces of information, other than news ID or title, that may be more descriptive (e.g., the news story) or better reflect a reader's interests (e.g., topics, categories) than titles or IDs. Second, the focus in these methods is usually on the relevancy and not on the diversity aspect.

Some works consider the two-tower deep neural networks to learn representation from content features in language models (Chidambaram et al., 2018; Logeswaran and Lee, 2018). These two-tower models are also used in recommender systems to leverage content features on the item side (Yi et al., 2019). Nevertheless, these models are usually focused on the relevancy of retrieved items, which is appreciated, however, they are not used to address the diversity aspect. In this work, we use a two-tower architecture for an NRS and try to incorporate both accuracy (relevancy) and diversity of news items while making recommendations to the users.

Maximal marginal relevance (MMR) is a classical technique to increase the diversity of documents retrieved against a query in an information retrieval system. MRR is also used in recommender systems to include diversity during the re-ranking phase of recommended items (Ziegler et al., 2005).

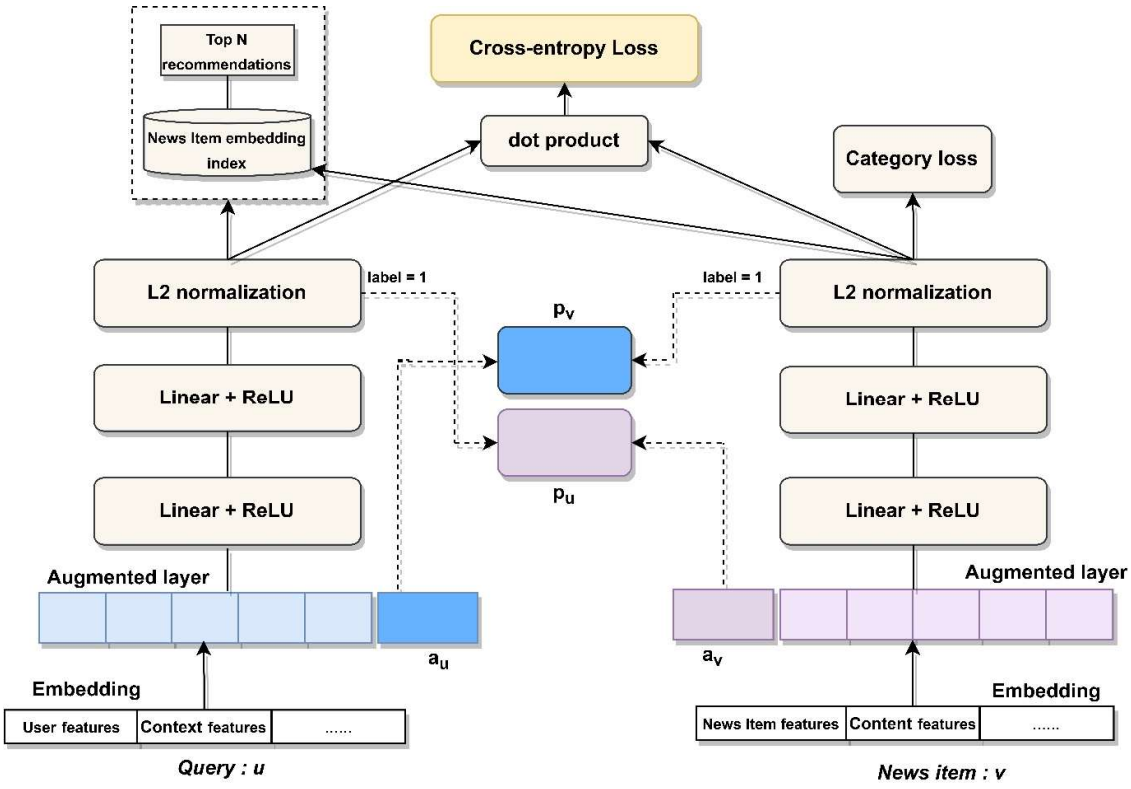

Figure 2: Network architecture for our proposed two-tower NRS

Some work (Raza and Ding, 2020; Qin and Zhu, 2013) considers diversity in the recommendation process by using the regularization terms on items' feature space. Dueling Bandit Gradient Descent (DBGD) (Afsar et al., 2021) is an online learning-to-rank algorithm based on multi-arm bandit algorithms and is used to model the exploration-versus-exploitation trade-off for relative feedback. DBGD is recently used in a state-of-the-art NRS (Zheng et al., 2018) to improve recommendation diversity. However, the learning efficiency of this model is limited in high-dimensional parameter space. Second, this method assumes only binary feedback because there is no way of directly observing the reward of users' actions. In this work, we also consider diversity during the model training time, but we use a category loss function for this purpose. Our intuition is that news items under different news categories are highly imbalanced and the recommendations are usually produced considering one such major category. We try to provide recommendations by considering news items from a diverse set of news categories.

3 Methodology

3.1 Problem Formulation

Considering the news item set $\{v_i\}_{i=1}^{N}$ and a query set as $\{u_j\}_{j=1}^{M}$, where N represents the number of news items, and M is the number of users, we refer to the news recommendation problem R, as selecting the candidate news items from the entire news corpus given a certain query.

We refer to a query as the feedback given by the user. We present the query-item feedback as a matrix $R \in \mathbb{R}^{N \times M}$. If the query j gives positive feedback on a news item i, then we consider it as $R_{ij} = 1$ (positive feedback), otherwise $R_{ij} = 0$. We represent each news item by the news content features, including news ID and side information, such as news title, body, and category. We also represent a user query with contextual factors, such as the time, and place when the user interacts with the item. By providing more information related to the query and the item, we can model rich interactions between two towers in a two-tower architecture.

3.2 Proposed two-tower architecture

We show our proposed framework for the two-tower architecture in Figure 2 and explain its work next:

Embedding Layer: The first layer is the embedding layer in the two-tower architecture. We define an embedding matrix E $\in \mathbb{R}^{K\times D}$, where E is the embedding matrix, K is the embedding dimension and D refers to the number of unique features. Each piece of information or feature in u_j and v_i goes through the embedding layer and is mapped to a low-dimensional dense vector, $e_j \in \mathbb{R}^K$, where e_j is the j^{th} column of E.

Augmented Layer: First, we create two input feature vectors $\mathbb{z}_u$ and $\mathbb{z}_v$ that contains the information about the current query vector u and the news item vector v. Then, we create two augmented vectors a_u and a_v by the IDs, corresponding to u and v respectively. These vectors a_u and a_v are then concatenated with $\mathbb{z}_u$ and $\mathbb{z}_v$ vectors to obtain the information from the feature vectors. We show $\mathbb{z}_u$ and $\mathbb{z}_v$ as shown in Equations (1) and (2) respectively:

$$\mathbb{z}_u = [e_{123} \; ||e_{To}||e_{time}|| \dots ||a_u] \quad (1)$$

$$\mathbb{z}_v = [e_{345} \; ||e_{sport}||e_{NYTimes}|| \dots ||a_v] \quad (2)$$

Where e_f, f is a feature that is related to u (e.g., place when the user interacts with the item) or v (e.g., news category sports or source NYTimes related to a news item), and the notation || refers to the vector concatenation operation.

The concatenated vectors $\mathbb{z}_u$ and $\mathbb{z}_v$ are fed into two towers (query and news item), which consist of the fully connected layers with the ReLU activation function. These layers receive the information between two towers through the augmented vectors a_u and a_v (a_u and a_v provides information about users' positive interactions). The output from the fully connected layers goes through ℓ_2 normalization layer that gives the augmented representations of query q_u and news item p_v. We define these steps formally as shown in Equation (3):

$$p = \ell_2 norm(h_L) \qquad (3)$$

$$h_L = \text{RELU}(W_l h_{L-1} + b_l)$$

$$h_1 = \text{RELU}(W_1 z + b_1)$$

where notation ℓ refers to loss, $\ell_2 norm$ is ℓ_2 −normalization, subscript L in h refers to layer, h is the intermediate representation, z denotes z_u and z_v , p refers to p_u and p_v ; W_l denotes the weight matrix in l^{th} layer, b refers to the bias vector. The representation p, is the output vector of the ℓ_2 normalization layer.

Loss function: We define the loss function as the mean square error between the augmented vector and query/item embedding for each sample of which label equals 1. The goal of the loss function is to use the augmented vector to fit all positive interactions in the tower belonging to the corresponding query/ news item. Formally, the loss functions are defined in Equations (4) and (5).

$$\ell_u = \frac{1}{T}\sum_{(u,v,y)\in Tr} [ya_u + (1-y)\, p_u - p_u]^2 (4)$$

$$\ell_v = \frac{1}{T}\sum_{(u,v,y)\in Tr} [ya_v + (1-y)p_v - p_v]^2 \quad (5)$$

where ℓ_u refers to the loss function with query vector and ℓ_v is a loss function associated with a news item vector. Tr is a training dataset, T refers to query-item pairs Tr, $y \in \{0,1\}$ is the label. If the label $y = 1$, it means the augmented vectors a_u and a_v approach the query embedding p_u and the news item embedding p_v, otherwise $y = 0$. We apply the stop gradient strategy to stop the gradient of ℓ_u and ℓ_v from flowing back into p_u and p_v respectively.

Once the augmented vectors a_u and a_v are obtained, they can model the information interaction between the two towers, and these vectors are considered as the input feature of the two towers. Finally, the output of the model is the inner product of the query embedding and news item embeddings, as shown in Equation (6):

$$s(u,v) =< p_u, p_v > \quad (6)$$

where $s(u,v)$ refers to the score provided by the proposed model.

News categories and diversity: In real-time news recommendation scenarios, the categories of news items are usually diverse (e.g., sports, politics, entertainment and so) and the number of news items under each new category is usually uneven. To incorporate diversity, we need to consider the recommendations from diverse news categories. To accomplish this, we propose another loss function related to the news category during the training phase that transfers the knowledge learned in one news category to the other news categories.

In particular, the news item representation p_v of a major news category $Cate^{major} = \{{p_v}^{major}\}$ (having the largest amount of data) is taken and transferred to other category sets

$Cate^2, Cate^3, Cate^4, ..$ and so. We define a loss function as the distance between the second-order statistics (covariances) of the major category and other news categories' features (Bello et al., 2008), shown in Equation (7):

$$\ell_{cate} = \sum_{i=2}^{n} ||\mathbb{C}\,(Cate^{major} - \mathbb{C}(Cate^{i})||_F^2) \quad (7)$$

where $\mathbb{C}\,(\cdot)$ denotes the covariance matrix, $||.||_F^2$ refers to the square matrix Frobenius norm and n is the number of news categories.

3.3 Model Training

We treat the news recommendation problem as a binary classification problem and use a random negative sampling technique, following the standard practice (Wu et al., 2020; Wu et al., 2021a; An et al., 2019) in most NRS. In particular, for each query in the positive query-item pair (the label = 1), we randomly sample N news items from the news corpus to create $\mathbb{S}$ negative query-item pairs (label = 0) with this query and add these $\mathbb{S}+1$ pairs to the training dataset. In the training process, we use binary cross-entropy to calculate the loss for the pairs, as shown in equation (8):

$$\ell_p = \frac{1}{T}\sum_{(u,v,y)\in Tr} [y log\sigma(\prec p_u, p_v \succ) + (1-y)\log(1-\sigma(\prec p_u, p_v \succ))] \quad (8)$$

$$Tr = D \times (\mathbb{S}+1)$$

where $\sigma\,(\cdot)$ refers to the sigmoid function, D denotes the number of positive feedback query-item pairs and Tr refers to the total number of training pairs.

Final loss function: The final loss function is calculated as shown in Equation (9):

$$\ell_{total} = \ell_p + \gamma_1\ell_u + \gamma_2\ell_v + \gamma_3\ell_{Cate} \quad (9)$$

where $\gamma_1, \gamma_2, \gamma_3$ refers to the tunable parameters.

4 Experiment

4.1 Datasets

MIcrosoft News Dataset (MIND): MIND (Wu et al., 2020) is a benchmark dataset consisting of anonymized behaviour logs from Microsoft News.

New York Times (NYTimes): we collected the news articles and the anonymized readers' interactions using NYTimes API. A sample of the dataset can be accessed here[1]. Both datasets consist of English news articles. We generated the training samples from the click histories and impression logs according to the format given in the MIND paper (Wu et al., 2020). The basic details for both datasets are shown in Table 1.

Dataset	MIND -small	NYTimes
Duration	6 weeks (12th Oct. 2019 - 22nd Nov. 2019)	2 years (1st Jan. 2017 - 31st Dec. 2018)
Readers	50,000	240,000
News	161,013	15,000
Clicks	156,925	2,000,000
News information	ID, headline, snippet (abstract), category, subcategory, publication timestamp	
Reader information	ID, interaction timestamp, click history, impressions	

Table 1: Dataset details

4.2 Evaluation methodology and metrics

Following the standard evaluation methodology and metrics in NRS (Wu et al., 2020; Wang et al., 2018), we conduct a time-based splitting and use the following evaluation metrics.

Accuracy metrics: Normalized Discounted Cumulative Gain (NDCG) and F1-score (harmonic mean of precision and recall).

Diversity metric: we use GiniIndex (GINI) (Sun et al., 2019b) for diversity, as it is a commonly used diversity metric in this kind of problem like diversity and fairness (Wu et al., 2021b).

tradeoff (Raza and Ding, 2020; Raza and Ding, 2021a)*:* We consider the trade-off between the mean F1-score (accuracy) and mean GINI index (diversity) scores. Our trade-off metric is defined in Equation 10:

$$tradeoff = 2 * \frac{(accuracy * diversity)}{(accuracy + diversity)} \quad (10)$$

Both the F1 and GINI scores are in the range [0,1]. We take the mean of top @ 10, 20 and 50 for calculating means of accuracy and diversity. We show the results in percentages.

4.3 Baselines Methods

We use the following baseline methods:

BERT4Rec (Sun et al., 2019a) with bidirectional self-attention to model user behaviour sequences. We use Bayesian Personalized Ranking (BPR) (Rendle et al., 2012) as the loss function.

[1] https://github.com/deeplearningnrs/D2NN

		MIND			NYTimes		
Model	**k**	**NDCG**	**F1**	**Gini**	**NDCG**	**F1**	**Gini**
Our approach	10	43.80%	36.90%	**87.40%**	51.00%	38.00%	**78.30%**
	20	50.10%	44.60%	*82.70%*	53.50%	45.00%	73.90%
	50	**63.50%**	**62.10%**	77.40%	**66.70%**	**65.50%**	73.30%
	trade-off	**68.91%**			**69.18%**		
BERT4Rec	10	41.80%	24.50%	74.30%	43.00%	25.00%	74.00%
	20	56.60%	42.20%	72.40%	58.00%	42.50%	71.80%
	50	*60.10%*	*57.90%*	70.60%	*62.00%*	*60.10%*	69.00%
	trade-off	*63.62%*			*64.24%*		
Two-tower	10	37.20%	44.50%	60.00%	37.80%	45.20%	56.00%
	20	38.10%	48.50%	57.90%	39.60%	49.50%	52.50%
	50	39.10%	49.70%	54.50%	42.00%	51.20%	51.30%
	trade-off	51.99%			51.25%		
Youtube-DNN	10	31.40%	35.30%	69.40%	33.00%	36.10%	65.00%
	20	34.50%	39.40%	58.90%	35.20%	40.40%	55.70%
	50	41.40%	46.40%	57.90%	42.00%	47.00%	54.20%
	trade-off	51.52%			50.34%		
LSTUR	10	39.70%	30.70%	81.20%	35.80%	22.10%	*76.40%*
	20	44.90%	32.60%	79.70%	42.00%	28.90%	75.90%
	50	55.10%	49.30%	72.50%	52.40%	34.20%	69.90%
	trade-off	58.69%			45.93%		
MultiVAE	10	32.90%	32.00%	58.90%	38.90%	34.20%	53.40%
	20	42.30%	39.00%	58.50%	44.00%	40.20%	51.10%
	50	43.40%	42.60%	57.60%	45.70%	44.50%	50.00%
	trade-off	48.98%			47.09%		
RepeatNet	10	37.80%	22.00%	74.70%	38.00%	25.00%	74.30%
	20	39.90%	24.10%	69.20%	41.00%	24.50%	65.80%
	50	43.20%	40.80%	65.30%	44.50%	41.00%	62.50%
	trade-off	50.22%			49.52%		
SASRecF	10	31.70%	24.00%	77.50%	32.00%	24.40%	74.00%
	20	32.70%	31.80%	75.80%	33.50%	32.00%	70.00%
	50	35.40%	32.10%	72.20%	36.80%	33.40%	68.80%
	trade-off	44.44%			44.97%		
ENMF	10	27.20%	14.50%	64.60%	28.90%	18.30%	63.70%
	20	28.10%	23.50%	57.90%	31.20%	25.00%	55.00%
	50	31.40%	29.70%	54.50%	33.40%	32.40%	52.40%
	trade-off	38.45%			40.04%		

Table 2: Performance of all models (Bold means best result, italic is second best, and underline is third-best)

MultiVAE (Liang et al., 2018), a collaborative filtering method with variational autoencoders. We use the cross-entropy as the loss function type.

ENMF (Chen et al., 2020) is an efficient matrix factorization method without sampling. We use the cross-entropy as the loss function for this model.

SASRecF (Zhang et al., 2019) is a feature-level self-attention model. We choose the BPR as the loss function.

RepeatNet (Ren et al., 2019) is a session-based recommender. We choose the BPR as loss function.

LSTUR (An et al., 2019) is an NRS that addresses long-short term users' preferences. We minimize the summation of negative log-likelihood of all positive samples during training,

Two-tower Model (Huang et al., 2013), is a standard two-tower model in retrieval tasks to leverage rich content features.

YouTubeDNN (Covington et al., 2016) is also a two-tower approach that feeds vectors into a multi-layer feed-forward neural network.

4.4 Hyperparameters

We implemented these models in TensorFlow. The embedding dimension and batch size were fixed to 32 and 256. We use the Adam optimizer. Other hyperparameters of all models were individually tuned to achieve optimal results to ensure a fair comparison. The dimensions of augmented vectors a_u and a_v were both set to $d = 32$, the tuning

parameter γ_1, γ_2 were set to 0.5 and γ_3 to 1. We set top@ k to 10, 20 and 50, as it is normally good practice to retrieve a relatively large number of candidate news items to rank.

5 Results and Analysis

5.1 Overall results

The comparison between our model and baselines is shown in Table 2. These scores are calculated using top@ 10, 20 and 50. We expect a good tradeoff score to be above 50% as it is a harmonic mean score.

Overall, we see in Table 2 that our proposed model has the highest accuracy, diversity and trade-off values on both datasets (MIND and NYTimes). This is shown by the highest F1-score, NDCG, Gini index and trade-off values of our approach among all the baseline methods. The superiority of our model is attributed to its design which has the following properties: 1) it considers the rich side information from the news content, 2) it considers the contexts in users' queries to better capture the sequential patterns of users' clicks, and 3) it considers news item representation of uneven categories and provide diversified recommendations.

Among baselines, the general performance of BERT4Rec is better than other baselines. BERT4Rec has also shown good performance in the general recommendation tasks through bi-directional contexts. The accuracy and tradeoff scores of BERT4Rec are also quite high.

Next, comes the performance of two-tower and Youtube-DNN methods in terms of tradeoff scores, both of which are based on two-tower architecture. The two-tower model performs better than Youtube-DNN in most scores. These methods provide the advantage of feature interactions, so a higher accuracy from these models was also expected. However, the diversity scores of these models are around 50-60%, which is not too high, probably because, they do not consider the diversity from uneven categories as we are incorporating into our approach.

The performance of LSTUR, in terms of tradeoff score, comes next. This model is constructed from the start to model news and user-specific information in their default configuration. In other models, incorporating news and user-modelling information may be mandated. LSTUR performs better in the MIND dataset, which is the default dataset (An et al., 2019) in the original paper. Compared to the accuracy scores of LSTUR, we see good diversity scores from this model (after our proposed model). The MIND dataset considers session-based information, so some diversity scores from LSTUR is expected on this dataset. LSTUR also shows good diversity in NYTimes. LSTUR also performs better than Youtube-DNN on MIND dataset.

MultiVAE is a collaborative filtering system and a non-linear probabilistic model. In terms of accuracy, the MultiVAE performs average in our experiments, and the model's diversity scores are not quite high. This is most likely because CF models (such as MultiVAE) mainly focus on the personalized recommendation strategy (Su and Khoshgoftaar, 2009), which identifies similarities between users/items to serve relevant product recommendations. As a result, we can expect the model to provide some accurate recommendations, but not too high diversity.

According to our results, the session-based recommenders i.e., SASRecF, and RepeatNet all have low-to-moderate accuracy. however, we see that these models show some higher diversity (above 50%). When it comes to providing diverse recommendations, usually the session-based recommender systems perform well (Karatzoglou and Hidasi, 2017). This is probably because of the ability of these models to recommend new and less similar items that users interacted within a session.

ENMF, a collaborative filtering method, has low-to-average accuracy and diversity scores resulting in average trade-off scores in our experiments. This suggests that we should probably extend a typical model to include more contexts, sequential information, or other recommendation models in order to provide better recommendations to users.

Overall, we see better results with the MIND dataset compared to NYTimes dataset. Due to brevity concerns and better overall results with the MIND dataset, we present the results of the subsequent experiments using the MIND dataset.

In the later experiments, we are only reporting the results on MIND dataset based on better results of all models on this dataset.

5.2 Accuracy-diversity trade-off

In this section, we showcase the accuracy-diversity trade-off achieved by our model. Figure 3 shows that as accuracy (mean F1-score) increases,

diversity (GINI) decreases, indicating an inherent relationship between these two evaluation aspects, which has also been validated in previous research (Adomavicius and Kwon, 2008; Raza and Ding, 2020; Raza and Ding, 2021a; Isufi et al., 2021).

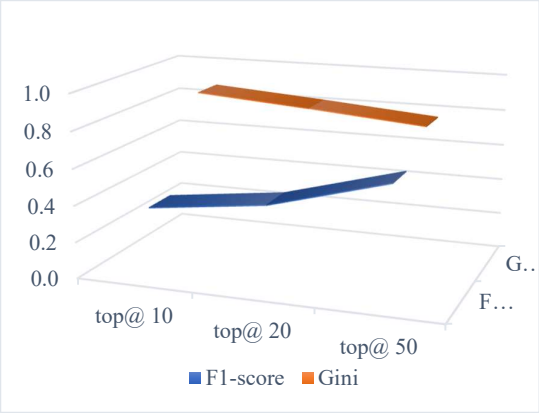

Figure 3: Accuracy and diversity trade-off of our model

As shown in Table 2, our model has the highest overall accuracy and diversity among all baselines, which is supported by a balanced trade-off score. Figure 3 also shows that as we increase the length of the recommendation list (top@ k), the accuracy of our model increases, whereas the diversity decreases. This increase in accuracy is due to the recall, which increases as the recommended items increase.

We also test the effectiveness of using different evaluation modes, which are discussed next:

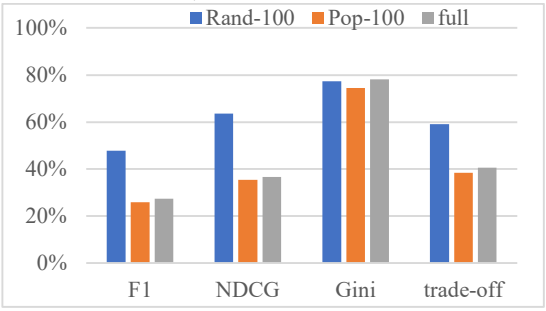

Figure 4: Model performance using different modes

random X (rand-X): randomly sample *X* negative items for each positive item in the testing set. *popularity X (pop-X):* sample *X* negative items for each positive item in the testing set based on item popularity. *full ranking*: evaluating the model on item sets. Rand-100 and pop-100 are negative sampling techniques. We report the results with *X*=100 based on best results, on the average of top @k (10, 20 and 50) and show scores in Figure 4.

Figure 4 shows that the rand-100 has the highest accuracy (F1, NDCG) score. We also see that the rand-100 mode gives us the highest diversity score (GINI-index). The pop-100 evaluation mode considers the 100 negative items for each positive item in the testing set based on item popularity and shows good results too, this is much like a collaborative filtering scenario, where some diversity is often compromised (Boim et al., 2011). The model variant with full evaluation mode shows good accuracy after the rand-100 variant. Overall, we find that the random negative sampling technique is a useful evaluation technique to achieve a balance between accuracy and diversity.

5.3 Effectiveness of side information

We also test the effectiveness of our model with and without the news side information (news body, news category, title). we consider the news body as a piece of side information in this experiment. The results are shown below in Table 3:

Metric	All features	Without news body
F1@10	36.90%	25.90%
F1@20	44.60%	21.40%
F1@50	62.10%	15.20%
Gini@10	87.40%	77.40%
Gini@20	82.70%	72.70%
Gini@50	77.40%	69.20%
tradeoff	**60.50%**	32.40%

Table 3. Model performance for side information.

As can be seen in Table 3, more content features improve the model performance compared to when we do not consider the news body. This signifies the importance of including more content-based features in the item tower that will interact with the query tower. Similarly, including more contextual features also improves model performance, we could not show this result due to limited space.

6 Conclusion

This paper proposes a two-tower architecture to model the information interaction between the two towers (query and news items). Extensive experiments on two datasets show the better performance of the proposed approach in achieving a balance between accuracy and diversity. In future, we like to conduct experiments on more real-world news data and make a deeper neural network. We like to evaluate our approach using more diversity metrics, such as normalized topic coverage and novelty. We also like to include more users' feedback like click-through rate.

References

Gediminas Adomavicius and Young Ok Kwon. 2008. Overcoming accuracy-diversity tradeoff in recommender systems: A variance-based approach. In *2008 Workshop on Information Technologies and Systems, WITS 2008*, pages 151–156. arXiv, September.

M Mehdi Afsar, Trafford Crump, Behrouz Far, M Mehdi Afsar, Trafford Crump, and Behrouz Far. 2021. Reinforcement learning based recommender systems: A survey. *arXiv preprint arXiv:2101.06286*.

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 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 336–345.

Juan Pablo Bello, Elaine Chew, and Douglas Turnbull. 2008. *ISMIR 2008: Proceedings of the 9th International Conference of Music Information Retrieval*. Lulu. com.

Rubi Boim, Tova Milo, and Slava Novgorodov. 2011. Diversification and refinement in collaborative filtering recommender. *International Conference on Information and Knowledge Management, Proceedings*:739–744.

Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient neural matrix factorization without sampling for recommendation. *ACM Transactions on Information Systems (TOIS)*, 38(2):1–28.

Laming Chen, Guoxin Zhang, and Hanning Zhou. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 5627–5638.

Muthuraman Chidambaram, Yinfei Yang, Daniel Cer, Steve Yuan, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning cross-lingual sentence representations via a multi-task dual-encoder model. *arXiv preprint arXiv:1810.12836*.

Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198.

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 *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Elvin Isufi, Matteo Pocchiari, and Alan Hanjalic. 2021. Accuracy-diversity trade-off in recommender systems via graph convolutions. *Information Processing and Management*, 58(2):102459, March.

Alexandros Karatzoglou and Balázs Hidasi. 2017. Deep learning for recommender systems. In *RecSys 2017 - Proceedings of the 11th ACM Conference on Recommender Systems*, pages 396–397. June.

Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, pages 689–698.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*.

Soroush Ojagh, Mohammad Reza Malek, Sara Saeedi, and Steve Liang. 2020. A location-based orientation-aware recommender system using IoT smart devices and Social Networks. *Future Generation Computer Systems*, 108:97–118.

Lijing Qin and Xiaoyan Zhu. 2013. Promoting diversity in recommendation by entropy regularizer. *IJCAI International Joint Conference on Artificial Intelligence*:2698–2704.

Shaina Raza and Chen Ding. 2020. A Regularized Model to Trade-off between Accuracy and Diversity in a News Recommender System. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 551–560.

Shaina Raza and Chen Ding. 2021a. Deep Neural Network to Tradeoff between Accuracy and Diversity in a News Recommender System. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 5246–5256.

Shaina Raza and Chen Ding. 2021b. News recommender system: a review of recent progress, challenges, and opportunities. *Artificial Intelligence Review*:1–52.

Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2019. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4806–4813.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 2009:1–19.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019a. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1441–1450, New York, NY, USA. Association for Computing Machinery.

Wenlong Sun, Sami Khenissi, Olfa Nasraoui, and Patrick Shafto. 2019b. Debiasing the human-recommender system feedback loop in collaborative filtering. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 645–651.

Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *The Web Conference 2018 - Proceedings of the World Wide Web Conference, WWW 2018*, pages 1835–1844, New York, New York, USA, April. Association for Computing Machinery, Inc.

Ruoxi Wang, Zhe Zhao, Xinyang Yi, Ji Yang, Derek Zhiyuan Cheng, Lichan Hong, Steve Tjoa, Jieqi Kang, Evan Ettinger, and Ed H Chi. 2019. Improving Relevance Prediction with Transfer Learning in Large-scale Retrieval Systems.

WeAreNetflix. 2018. Netflix Research: Analytics. (Accessed on 02/04/2022).

Chuhan Wu, Fangzhao Wu, Yang Yu, Tao Qi, Yongfeng Huang, and Xing Xie. 2021a. UserBERT: Contrastive User Model Pre-training.

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, Stroudsburg, PA, USA, July. Association for Computational Linguistics.

Yao Wu, Jian Cao, and Guandong Xu. 2021b. Fairness in Recommender Systems : Evaluation Approaches Fairness in Recommender Systems : Evaluation Approaches and Assurance Strategies.

Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. 2020. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion Proceedings of the Web Conference 2020*, pages 441–447.

Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *RecSys 2019 - 13th ACM Conference on Recommender Systems*, pages 269–277.

Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level deeper self-attention network for sequential recommendation. In *IJCAI International Joint Conference on Artificial Intelligence*, volumes 2019-Augus, pages 4320–4326.

Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *The Web Conference 2018 - Proceedings of the World Wide Web Conference, WWW 2018*, pages 167–176, New York, New York, USA, April. Association for Computing Machinery, Inc.

Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. 2019. DAN: Deep attention neural network for news recommendation. In *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, volume 33, pages 5973–5980.

Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, number January, pages 22–32.