Towards Enhancing Multi-task Learning for News Recommendation

STEFAN VASILEV, University of Amsterdam, Netherlands
MATEY KRASTEV, University of Amsterdam, Netherlands
DANILO TOAPANTA, University of Amsterdam, Netherlands

In this report we document our work on reproducing and adapting MTRec [2], a news recommendation method using pre-trained BERT, for the 2024th edition of the RecSys challenge. MTRec extracts a user representation vector from clicked user articles and scores candidate articles for recommendation by calculating a dot product with each candidate article's representation. The authors posit auxiliary tasks to aid learning and propose the use of gradient surgery to combine the main task and the auxiliary gradients to the respective losses. In this research, we explore a different auxiliary task, i.e sentiment classification to aid the learning of our task. We further propose to use LoRA [6] instead of full fine-tuning, which we later show to have a regularizing effect and to yield a slightly better performing model than the original authors' model. Our ablations verify the validity and importance of the included methodological choices.

CCS Concepts: • **Information systems** → **Recommender systems**; **Information retrieval**; • **Computing methodologies** → **Information extraction**.

Additional Key Words and Phrases: Recommendation, News, Multi-task Learning, BERT

ACM Reference Format:
Stefan Vasilev, Matey Krastev, and Danilo Toapanta. 2025. Towards Enhancing Multi-task Learning for News Recommendation. 1, 1 (September 2025), 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Recommendation tasks permeate and consistently interplay with real-world scenarios designed to enhance user experience, boost user engagement, and improve the relevancy of suggestions for the potential user. Contextualized within news recommendation, the task entails providing users with news articles suited to the individual user's preferences based on a scoring mechanism between the user's and the article's latent representations.

As a universal means of learning latent representations of information, deep neural network-related methodologies have been proven to significantly outperform all other alternatives, and in particular for NLP tasks [5, 8]. These foundation models, however, are initially trained as generalists and are intended to be fine-tuned on task-specific data or target domains. Specifically, for news recommendation, the task entails encoding information for a news article, usually from the title, body or some other combination of the article content, and then applying the learned representations for downstream tasks such as learning optimal user representations [2, 9] and user-article matching based on some pre-collected preference data, such as click rate or popularity.

Authors' Contact Information: Stefan Vasilev, stefan.vasilev@student.uva.nl, University of Amsterdam, Netherlands; Matey Krastev, University of Amsterdam, Netherlands, matey.krastev@student.uva.nl; Danilo Toapanta, University of Amsterdam, Netherlands, danilo.toapanta@student.uva.nl.

ACM XXXX-XXXX/2025/9-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

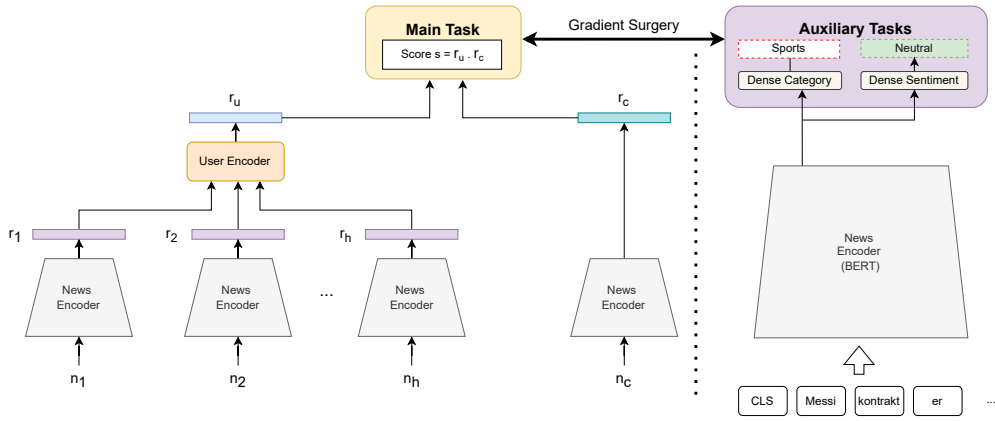

Fig. 1. **MTRec Model Training Architecture [2].** History news titles are fed to the user encoder, which aggregates them to a user representation vector r^u. A main task score s is calculated between an r^u and a candidate news vector r^c. An auxiliary news category and sentiment classification losses are calculated for each news piece in the history. Gradient surgery is used to combine the main task and the auxiliary losses to aid the main task loss.

We examine the MTRec [2] framework for multi-task learning over BERT for news recommendation, by adapting it to a different dataset and proposing several extensions to improve the robustness and accuracy of the original authors' method, achieving an improvement of around 0.5 AUC score.

2 Methodology

In this section, we highlight the key concepts introduced by the original authors, their contribution to the field, as well as our own proposed extensions.

2.1 Problem Formulation

Item recommendation can be based on a pre-recorded user session of clicked articles, either in some specific interval or across the user's entire history. Thus, given the set of I previously clicked articles $N^h = [n_1^h, n_2^h, ..., n_I^h]$, and another set of J candidate news $N^c = [n_1^c, n_2^c, ..., n_J^c]$, our goal is to calculate the user interest score s_j of each candidate news according to the historical behavior of the user. This score can then be used as-is for ranking or combined with other features in order to provide appropriately ranked news recommendations.

Furthermore, we assume each article to be characterized by several identifying features, namely, its title text t, category label $c \in C^K$, and entity set ϵ of named entities in the title. Additionally, other features may be available, such as body text, attached images, or sentiment, among others.

2.2 Main Task

Multi-task learning aims to enhance the performance of a target main task by introducing auxiliary tasks that are learned jointly along with the main task. Our main task is learning user interest scores s_j based on the user's historical interests and the representations for the candidate articles.

2.2.1 News Encoder. The news encoder takes in the identifying features of an article and encodes them into an aggregate representation. Following the original authors [2], we adapt a pre-trained BERT model [5] for this task and only use the title text t of a given news article. Then, we take the last hidden state of the added [CLS] token as the concrete representation of the article. Following

the original authors' convention, we denote the encoded vectors of historical clicked news N^h and candidate news N^c as $R^h = [\mathbf{r}_1^h, \mathbf{r}_2^h, \ldots, \mathbf{r}_I^h]$ and $R^c = [\mathbf{r}_1^c, \mathbf{r}_2^c, \ldots, \mathbf{r}_J^c]$, respectively.

2.2.2 User Encoder. The user encoder aggregates the representations of the historical clicked news, and existing methods usually employ sequential [1] or attentive models [9]. For their specific purposes, the authors employ the following additive attention-based encoder, yielding the user embedding $\mathbf{r}^u$:

$$\mathbf{r}^u = \sum_{i=1}^{I} \mathbf{a}_i^u \odot \mathbf{r}_i^h, \quad \mathbf{a}_i^u = \text{softmax}(\mathbf{q}^u \odot \tanh(\mathbf{W}^u \cdot \mathbf{r}_i^h)), \tag{1}$$

where $\mathbf{W}^u$ and $\mathbf{q}^u$ are a trainable parameter matrix and vector, respectively; $\odot$ denotes the element-wise multiplication operator; and the tanh function is also applied element-wise.

2.2.3 Scoring. For each candidate news vector $\mathbf{r}_j^c$, and an associated user embedding $\mathbf{r}_j^u$, we calculate the click predictor score s_j via their dot product:

$$s_j = \mathbf{r}_j^u \cdot \mathbf{r}_j^c \tag{2}$$

2.3 Multi-task Learning

The authors note that, beside the content of the article (title, subtitle, body, etc.), other data, such as category, named entities, etc., is usually available and can be utilized as additional input features augmenting the score prediction capabilities of the model. However, these additional features may not be trivially incorporated within the deep representations of BERT and therefore cause ineffective use of this multi-field information.

Instead, the authors propose propose to train the model jointly on both the main task, as well as several auxiliary tasks that can potentially enable it to adapt to the target language domain better and thus enhance the model's capabilities, namely category classification and name entity recognition.

Category Classification. Each article can be broadly associated with a specific category, e.g. sports, politics, crime, etc. Given an article set, where each article belongs to a set of K pre-defined categories K^C, the model will predict the probability $\hat{p}^k$ that every article can be classified in a category $k \in K^C$.

The original authors employ a classification head layer on top of the [CLS] token of a news article title n_i:

$$\hat{\mathbf{p}}_i^k = \text{softmax}(\mathbf{W}^k \mathbf{r}_i + \mathbf{b}^c) \tag{3}$$

where $\mathbf{W}^c$ and $\mathbf{b}^c$ are trainable parameters and the k-th element of the $\hat{\mathbf{p}}_i^c$ vector denotes the probability that the article n_i belongs to the k-th category. We employ one-hot encoding for the categories.

Named Entity Recognition. The authors employ Named Entity Recognition (NER) as one of the auxiliary tasks, however, this requires manually labeled samples of named entities in the dataset in order to ensure model accuracy. Moreover, it was found that the Ekstra Bladet News Recommendation Dataset contained entities extracted from the body of the article as well as the subtitle and title respectively. With the current implementation of MTRec which uses only the title to train NER classification it was found that there was nearly 90% missing NER cluster by only looking at the title as described by the original paper. For this reason it was considered this signal will not improve the model's performance and instead we prefer to diverge from their original

implementation in order to adopt a more widely-applicable approach for fine-tuning BERT on auxiliary tasks.

Sentiment Analysis. We diverge from the original paper by substituting NER with title-based sentiment classification. Similarly to the category encoder described above, we implement a K-class trainable sentiment classification head on top of the base model.

2.4 Loss Function

Following the original work [2], we employ the NCE loss for the main task loss $\mathcal{L}_{\text{MAIN}}$, where we contrast the scores of the positive clicked articles with L sampled negative non-clicked but in-view articles. The negative samples are drawn without replacement from a uniform distribution of the in-view articles, and we empirically set $L = 4$ following the original authors.

$$\mathcal{L}_{\text{MAIN}} = -\sum_{i=1}^{|D|} \log \frac{\exp(s_i^+)}{\exp(s_i^+) + \sum_{j=1}^{L} \exp(s_i^j)} \tag{4}$$

For the auxiliary tasks, we employ traditional cross-entropy loss over K classes.

$$\mathcal{L}_{\text{CAT}} = -\sum_{i=1}^{|D|} \sum_{j=1}^{I} \sum_{k}^{K} p_{ijk} \log \hat{p}_{ijk} \tag{5}$$

Naively combining the losses, we obtain the final model loss:

$$\mathcal{L}_{\text{MTRec}} = \mathcal{L}_{\text{MAIN}} + \mathcal{L}_{\text{CAT}} + \mathcal{L}_{\text{SENT}} \tag{6}$$

2.5 Gradient Surgery

We simultaneously train the base model (in this case, BERT) for the main task and the auxiliary tasks. However, the auxiliary tasks may be pointing in opposite directions with each other or the main task, in particular with larger than 90° angles (Figure 2a), therefore inhibiting effective learning for the main task. To address this, the original authors propose incorporating gradient surgery [10] in the training phase. Gradient Surgery projects the gradients of the tasks in multi-task learning to a shared plane in order to align them in the same overall direction and therefore achieve more effective training.

$$\mathbf{g}_{\text{MAIN}} = \mathbf{g}_{\text{MAIN}} - \frac{(\mathbf{g}_{\text{AUX}} \cdot \mathbf{g}_{\text{MAIN}})}{||\ \mathbf{g}_{\text{AUX}}\ ||^2} \cdot \mathbf{g}_{\text{AUX}} \qquad \mathbf{g}_{\text{AUX}} = \lambda(\mathbf{g}_{\text{CAT}} + \mathbf{g}_{\text{SENT}}) \tag{7}$$

In addition, the original authors [2] apply a scaling factor λ to the auxiliary loss which they empirically set to 0.3, in order to guide the gradients into prioritizing the main task loss during weight updates.

2.6 LoRA

LoRA [6] is a parameter efficient fine-tuning method which decomposes the weight updates into low-rank matrices, and has been shown to achieve identical performance to full fine-tuning, while also serving a regularizing function during fine-tuning, thus leading to more robust predictions [3]. We hypothesize that applying LoRA adapters to the layers of the base model may yield more stable training and less overfitting to the train set.

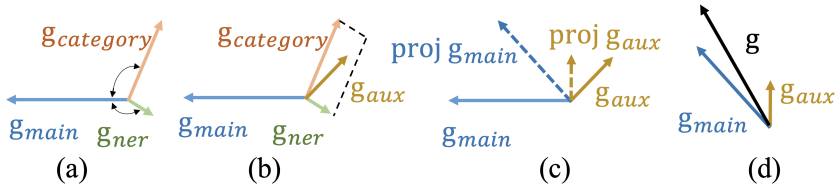

Fig. 2. **Illustration of Gradient Surgery.** Without GS (a), gradients will be pointing in different directions. Thus, we merge the gradients of the auxiliary tasks (b). Finally, we project gradients of both main and auxiliary tasks (c) and combine them (d).

3 Evaluation

3.1 Dataset

For training and evaluation, we utilize the Ekstra Bladet News Recommendation Dataset (EB-NeRD) [4], which contains user behaviour logs at the Danish news website Ekstra Bladet. Users are anonymized by IDs and each user has a history of clicked articles, session logs, which contain clicked articles and viewed-but-not-clicked articles in a given user session. A single user can have many sessions.

The EB-NeRD dataset contains the title, subtitle (or abstract) and body of the news article. An article is associated with category, attached images, and other background information. Finally, the dataset authors provide NER tags of the article titles and the article topical categories, produced by a proprietary tagging model.

The dataset comes in three subsets, ordered by smallest to largest: demo, small, and large, each containing a training and validation split. Additionally, the dataset authors provide a separate test split, similar in size to the large subset. Due to computational constraints, we choose to train our models on the demo subset for prototyping and the small subset for our results.

3.2 Metrics

We examine the experimental ranking results on several key metrics.

- **AUC:** Measures the model's ability to distinguish between clicked and non-clicked articles
- **MRR:** Estimates how well relevant articles are ranked in the top positions
- **nDCG@K:** Evaluates the ranking quality by considering the position of relevant items among the top K results

3.3 Models

- **BERT (Baseline):** Serves as our baseline, employing pre-trained BERT embeddings we fine-tune to the dataset, without any modifications or auxiliary tasks. This setup provides a benchmark for assessing the enhancements introduced by our proposed methods.
- **MTRec (OG):** Represents the original MTRec model that incorporates all proposed modifications and auxiliary tasks. The performance metrics of this configuration establish the effectiveness of our full model setup.
- **MTRec (Ours):** This is the adaptation of the MTRec model with the added improvements such as the sentiment classifier and LoRA-based fine-tuning to the EB-NeRD dataset.

Furthermore, we analyse the impact of our improvements by doing ablations on the different components we implement.

Method	AUC	MRR	nDCG@5	nDCG@10
BERT (Baseline)	50.13	-	20.64	20.17
MTRec (OG)	50.49	31.89	35.21	43.47
MTRec (Ours)	**50.92**	**32.21**	**35.64**	**43.97**
w/o LoRA	50.57	32.02	35.41	43.79
w/o CAT Loss	50.23	31.75	35.09	43.56
w/o SENT Loss	50.35	31.86	35.13	43.78

Table 1. MTRec Performance and Ablations on the EBNeRD dataset.

3.4 Results

In this section we show the performance of our model, compared to the original authors' setup, with additional ablations on the different components of our model's architecture.

In Table 1 we see that our model performs slightly better in all metrics than the authors (OG) model on our task, while both models are valid methods, outperforming the baseline significantly.

Our ablation without LoRA, i.e full fine-tuning, outperforms the original authors' model, however, falls under the LoRA version of the model, due to LoRA's regularising efect. One can further examine this effect in the loss curves in Figure 3. Overall, the loss is much smoother and steadily decreasing than the Full fine-tunining variant.

Our two other ablations, without the article topic and the sentiment classification auxiliary tasks respectively, both show worse performance than our method, which demonstrates that they are both important for learning the task.

4 Discussion and Conclusion

We observe that the auxiliary tasks result in a marginal increase of performance for the model. Quantitatively, the model is able to learn additional features of the target domain, both through the main task, as well as the auxiliary task.

The performed ablations highlight certain trends. In particular, the inclusion of LoRA has lead to the most significant improvement for the model over the baseline and the implementation of the original authors. LoRA not only improves training time, but also exhibits regularizing trends, as can be seen in Figure 3. The main task loss at later training iterations has less variance and is generally lower than when using full fine-tuning. Additionally, for the auxiliary tasks, the LoRA-fine-tuned model converges more slowly. We hypothesize that these properties indicate reduced overfitting and better generalization for the model to unseen data, indicating that LoRA fine-tuning acts as a regularizer, in line with the conclusions of [3]. Finally, we highlight that effective fine-tuning is key to good model performance.

Furthermore, substituting the original authors' implementation for NER auxiliary task with sentiment analysis task has proved effective to a certain extent, highlighting an additional potential venue for improvement during multi-task learning.

Overall, we demonstrate the effectiveness of MTRec in tackling the RecSys challenge for news recommendation and propose viable improvements for the task at hand, which are backed by ablations.

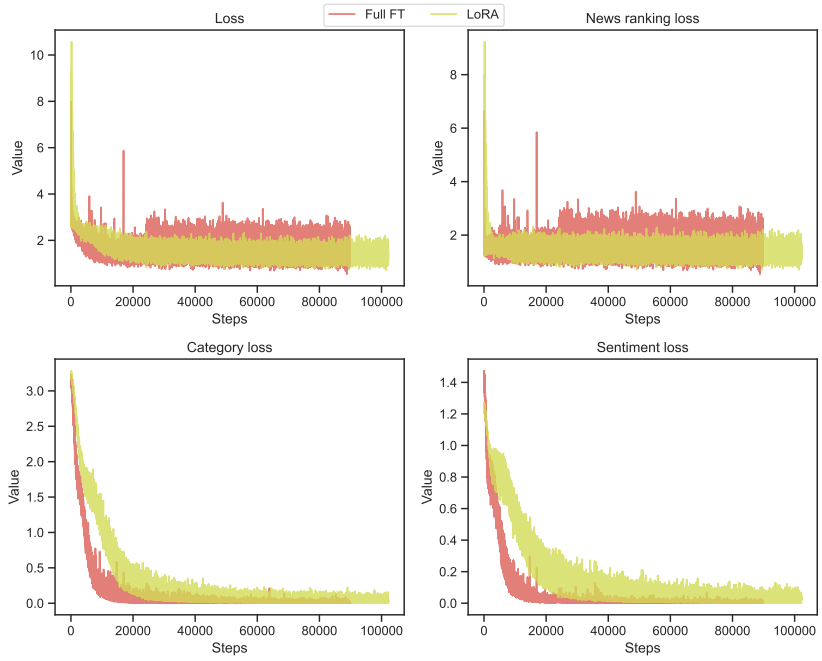

Fig. 3. Training curves for the full fine-tuning and the LoRA versions of the model.

5 Future Work

While the MTRec method shows promise, a lot of improvements could be made for better performance. The BERT backbone could be replaced with another better pre-trained model, such as RoBERTa [7]. Furthermore, the method only utilizes the news titles to extract representations, however, in the real world a lot more standard data is available for a news piece, such as, subtitle, body text, images/videos, article release date, etc. All of these data points could be utilized in a model to craft a fuller representation of a news article. Of utter importance to news recommendation is the recency of the article, which could be used as a feature, taking into account time of the user session and release date. Incorporating a notion of time into the model could give it the capability to reason about sequential patterns of user interest, thus making better news recommendations.

Ultimately, there is a myriad of directions to explore in improving MTRec. In that sense, the method is just scratching the surface of what is possible and sets a strong baseline for deep neural recommendation systems.

Acknowledgments

Many thanks to the organizers of the Ekstra-Bladet RecSys challenge for providing the exhaustive dataset, as well as for their thorough documentation and responsiveness throughout the challenge.

References

[1] 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 the 57th Annual Meeting of the Association for Computational Linguistics*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 336–345. https://doi.org/10.18653/v1/P19-1033

[2] Qiwei Bi, Jian Li, Lifeng Shang, Xin Jiang, Qun Liu, and Hanfang Yang. 2022. MTRec: Multi-Task Learning over BERT for News Recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 2663–2669. https://doi.org/10.18653/v1/2022.findings-acl.209

[3] Dan Biderman, Jose Gonzalez Ortiz, Jacob Portes, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. 2024. LoRA Learns Less and Forgets Less. https://doi.org/10.48550/arXiv.2405.09673 arXiv:2405.09673 [cs].

[4] Ekstra Bladet. 2024. Ekstra Bladet News Recommendation Dataset. https://recsys.eb.dk/dataset/ Accessed: 2024-06-23.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[6] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. https://doi.org/10.48550/arXiv.2106.09685 arXiv:2106.09685 [cs].

[7] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL] https://arxiv.org/abs/1907.11692

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. http://arxiv.org/abs/1706.03762 arXiv:1706.03762 [cs].

[9] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Multi-Head Self-Attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 6389–6394. https://doi.org/10.18653/v1/D19-1671

[10] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient Surgery for Multi-Task Learning. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 5824–5836. https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html

A Hyperparameters

We list all used hyperparameters and report the results of a single run, with a fixed seed of 42. The foundation for all models is a pre-trained BERT-base [5]. For all scenarios, we fine-tune on the small subset of the dataset for 10 epochs.

	Baseline	MTRec (OG)	MTRec (Ours)
Batch Size	16	16	16
Max Sequence Length	64	64	64
Learning Rate	2e-5	2e-5	2e-5
LoRA r	-	-	16
LoRA α	-	-	4
Optimizer	Adam	Adam	Adam
Weight Decay	1e-6	1e-6	1e-6
Warmup Steps	10%	10%	10%

Table 2. Hyperparameters.