Towards Advancing Multi-task Learning Over BERT for News Recommendation

Stefan Vasilev Matey Krastev Danilo Toapanta

University of Amsterdam

UNIVERSITY OF AMSTERDAM

Problem Definition

Item recommendation comprises computing user preference for candidate items. Assuming a pre-recorded user session of clicked articles 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.

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.

Multi-task learning aims to enhance the performance of a target model by introducing auxiliary tasks that are learned jointly along with the main task.

Contextualizing News Recommendation

Our main task is learning user interest scores s_j based on the user's historical interests and the representations for the candidate articles.

News Encoder

- The input to the news encoder is the **title of the news article** n
- We take the [CLS] token embedding from a pre-trained multilingual BERT as a representation of each article, following the original authors [1]

$$\mathbf{r}^h = \text{BERT}(n)[\text{CLS}] \tag{1}$$

User Encoder

- The user encoder aggregates the representations of the historically clicked news to create a model of each user.
- We employ the additive **attention-based encoder** [3], yielding the user embedding $\mathbf{r}^u$:

$$\mathbf{r}^u = \sum_{i=1}^{I} \mathbf{a}_i^u \odot \mathbf{r}_i^h, \quad \mathbf{a}_i^u = \text{softmax}(\mathbf{q}^u \odot \tanh(\mathbf{W}^u \cdot \mathbf{r}_i^h)), \tag{2}$$

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:

$$s_j = \mathbf{r}_j^u \cdot \mathbf{r}_j^c \tag{3}$$

References

[1] Q. Bi, J. Li, L. Shang, X. Jiang, Q. Liu, and H. Yang. MTRec: Multi-Task Learning over BERT for News Recommendation. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2663–2669, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[2] D. Biderman, J. G. Ortiz, J. Portes, M. Paul, P. Greengard, C. Jennings, D. King, S. Havens, V. Chiley, J. Frankle, C. Blakeney, and J. P. Cunningham. LoRA Learns Less and Forgets Less, May 2024. arXiv:2405.09673 [cs].

[3] C. Wu, F. Wu, S. Ge, T. Qi, Y. Huang, and X. Xie. Neural News Recommendation with Multi-Head Self-Attention. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6389–6394, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[4] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. Gradient Surgery for Multi-Task Learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc., 2020.

Methodology

- We only utilize **news titles as inputs**, instead of the full content of the article (body, subtitle, etc.), because they are most trivially processed by BERT to extract representations.
- Instead of using additional input data, such as category and named entities, the authors propose training the model to predict those features as **auxiliary tasks**. In our project, we experiment with **category classification and sentiment classification**.

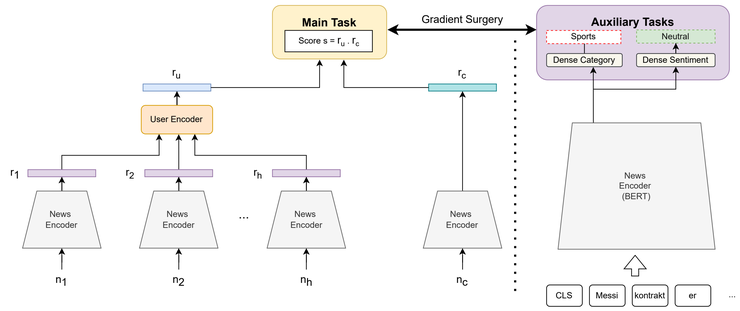

Figure 1. **MTRec Architecture [1]** History news titles are fed to the user encoder, which aggregates them to a user representation vector r_u. A main task score s is calculated between an r_u and a candidate news vector r_c. An auxiliary news category and sentiment classification losses are calculated 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.

Loss Functions

- Following the original work [1], 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 negative non-clicked articles in the same session.
- For the **auxiliary tasks**, we employ traditional **cross-entropy loss** over K classes.

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

(a) Main Task.

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

(b) Auxiliary Tasks

Gradient Surgery

- Gradients of different tasks may point in different directions [4]. In particular, directions forming an angle larger than 90° can harm each other.
- To alleviate this issue, we adapt Gradient Surgery with stress on the main task.

$$\mathbf{g}_{\text{MAIN}} = \mathbf{g}_{\text{MAIN}} - \frac{(\mathbf{g}_{\text{AUX}} \cdot \mathbf{g}_{\text{MAIN}})}{\| \mathbf{g}_{\text{AUX}} \|^2} \cdot \mathbf{g}_{\text{AUX}} \qquad \mathbf{g}_{\text{AUX}} = \lambda(\mathbf{g}_{\text{CAT}} + \mathbf{g}_{\text{SENT}}) \tag{4}$$

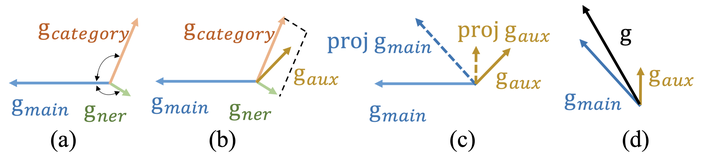

Figure 3. **Illustration of Gradient Surgery.** Without GS (a), gradients will be pointing in different directions. We merge the gradients of the auxiliary tasks (b), project the gradients of both main and auxiliary tasks (c), finally combining them (d).

Quantitative Results

To validate and re-affirm our assumptions as well as original ideas, 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

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.

Qualitative Results

- The model is able to effectively learn both the main task, as well as the auxiliary tasks.
- Finetuning with LoRA takes longer to converge but provides more robust results [2].

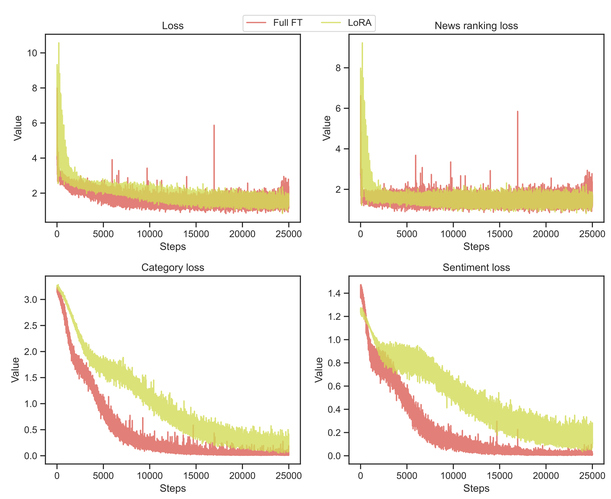

Figure 4. Training curves for the full fine-tuning and the LoRA versions of the model.

Future Work

- The user embeddings' robustness may be further improved and therefore suitable for *collaborative filtering*.
- Mine temporal data to provide recommendations that are more relevant to current trends.
- Inclusion of NER classification into the pipeline, i.e. entities from body, title or subtitle, as well as other auxiliary tasks.

Acknowledgements: Master AI at UvA

{stefan.vasilev;matey.krastev;danilo.toapanta}@student.uva.nl