LLM-Based Aspect Augmentations for Recommendation Systems

Reza Yousefi Maragheh [*1] **Lalitesh Morishetti** [*1] **Ramin Giahi** [1] **Kaushiki Nag** [1]
Jianpeng Xu [1] **Jaosn Cho** [1] **Evren Korpeoglu** [1] **Sushant Kumar** [1] **Kannan Achan** [1]

Abstract

Large language models (LLMs) have shown to be effective in different task settings, including recommendation-related tasks. In this study, we aim at measuring the effectiveness of using item aspects (justifications for users' intentions when buying the item) generated by LLMs in improving the results for ranking tasks. For this purpose, we carefully design prompts for LLMs to derive aspects for items using their textual data in an eCommerce setting. The extracted aspects are used as augmentations for Learning-to-Rank models. Specifically, we input the generated aspects as summarized embeddings using three approaches: (i) augmenting using feature concatenation, (ii) adding a wide aspect component beside a deep component of features, and (iii) adding an aspect embedding tower to create a two-tower model. We conduct extensive experiments on real-world eCommerce dataset and show the effectiveness of including LLM-based aspects in improving ranking metrics such as MRR and NDCG, even when they are compared to models augmented by pre-trained language models (PLM).

1. Introduction and Background

Conditional ranking task in eCommerce settings is referred to recommendation framework where there exist a set of items which act as the reference or conditions for recommendation (Hou et al., 2023b). For instance, in sequential recommendation setting, the user-item interaction sequence acts as the reference for recommendation (Yan et al., 2019; Song et al., 2021), or in item-pages of eCommerce, the main item of the page (see figure 1) acts as the reference for recommendation (Maragheh et al., 2022).

Figure 1. Conditional ranking task is one of the core tasks in eCommerce platforms. The recommendation module depicted in the picture aims at recommending items that are relevant to the main item of the page.

In these cases, deriving aspects or justifications for users' interests is crucial in improving recommendations. These justifications extract relevant keywords about the user behavior and aim at attaining the users' intentions (Ni et al., 2019). For instance, when parents are looking for toys for their children, they maybe interested in "educational" aspect of the toy, or its "durability". In this case, a recommendation system which is aware of the item aspects influencing user behavior will be more accurate in detecting the relevant items for the user.

However, most of the existing models for retrieval tasks are "narrow experts"(Guo et al., 2020), which are extremely domain oriented with task-specific objectives, and hence lack the capability of using common-sense universal knowledge such as aspects (Hou et al., 2023a).

Pre-trained Language Models (PLM) try to alleviate this issue by transferring the rich world knowledge from the universe of web textual data to better understand users' behavior and preferences (Hou et al., 2023a; 2022). Similarly, LLMs have shown excellent capabilities in common-sense reasoning and utilizing background knowledge in a variety of tasks.

Very recently, LLMs have shown promising results in recom-

[*]Equal contribution [1]Walmart Global Tech, Sunnyvale, CA, USA. Correspondence to: Reza Yousefi Maragheh <reza.yousefimaragheh@walmart.com>, Lalitesh Morishetti <lalitesh.morishetti@walmart.com>.

Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023.

mendation settings, specifically in zero-shot and few-shot learning for sequential recommendation (see Wang & Lim 2023; Gao et al. 2023; Wang et al. 2023; Zhang et al. 2023) and knowledge-graph completion (see Chen et al. 2023). Most of the work in this area focuses on carefully designing prompts by explicitly inputting examples and measuring the effectiveness of LLM's outputs in ranking or knowledge graph completion.

In this work, instead of using LLMs in zero-shot or few-shot ranking settings, we aim to use aspects generated by LLMs to improve the performance of existing Learing-to-Ranking (LtR) architectures. More specifically, using the associated textual data for each item, we generate aspects describing the items to better understand the intentions underlying the users' behavior for choosing that item. Then, we use these aspects to augment the inputs to LtR models. Our experiments on real-world data sets show that considering these aspects in some of the widely used state-of-the-art models can improve ranking metrics such as MRR and NDCG.

2. LLM-Based Aspect Generation

In this section, we review the aspect generation framework. Then, we show how the instructions are designed as prompt inputs to LLM, and how the aspects are parsed from the output.

2.1. Item Aspect Generation

In the item aspect generation, given the textual data corpora, we are interested in generating aspects why potential customers like an item using LLMs for extractive summarization. Formally, given an input of textual data for item i, $\mathcal{D}_i$, we are interested in generating k aspects $A_i = \{a_i^{(1)}, \ldots, a_i^{(k)}\}$ for users' potential interest in this item.

We generate the aspects using Google's PaLM2 (see Anil et al. 2023). To enable this LLM and to derive aspects for the items, we specially designed the prompt by including textual information and defining explicit response structure to the prompt. The textual information is composed by the concatenation of the item description and the user's positive reviews. Defining explicit response structure helps in output parsing and makes LLM understand the expected output structure. To this end, we start the prompt with phrase "*Summarize the following reviews in three adjectives. Reply in this format: relevant tags for this product are [first adjective, second adjective, third adjective]:*" and then input the concatenated description and reviews (See figure 2).

After generating all aspects for the items, we find the union of all the aspects as the aspects' universe $\mathcal{A}$. Then, we construct the item-aspect binary matrix. Depending on the

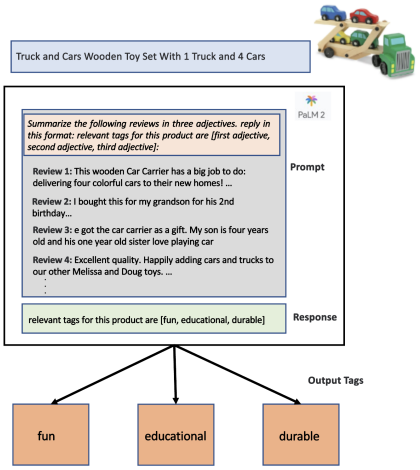

Figure 2. Desinged prompts for aspect generation using PaLM2.

generated number of aspects for each product, k, and the similarity of items, this matrix can be sparse. To generate aspect embeddings, we perform a dimensionality reduction using layer.embedding function of TensorFlow (Abadi et al., 2015). These reduced vectors will be inputted into the augmentation architectures of the ranking models.

2.2. Parsing the Output

The outputs of LLM are generated in an array format as designed in the prompt input. In our experiments, PaLM2 hallucinates for some of the prompts and starts the reply with irrelevant phrases. For instance, this is one of the generated responses by PaLM2: *so it will grow with her. The push bar is also a nice feature. Relevant tags for this product are [sturdy, durable, classic].* However, in these cases, the response ends with the expected array reply.

Also, in some cases and for some items, PaLM2 returns an empty string (approximately for 5% of the prompts). We consider these as errors and do not input any aspects of the items. Parsing is done by considering the expected array format for each output.

3. Models

As mentioned, the generated aspect embeddings are augmented with the item's features. We test three augmentation architectures to better understand how to use aspect embeddings in ranking tasks.

Note that this paper focuses on conditional ranking settings, where we are given an anchor item that acts as a recommendation reference and a candidate set of items selected accordingly. The conditional ranking task aims at ranking the more relevant items to the anchor/reference item in the

top positions.

3.1. LLM-Based Aspect Augmentation Models

We use three architectures to augment the generated aspect embeddings with other dense features/embeddings.

3.1.1. Embedding Concatenation

A baseline approach for augmenting the aspect embeddings with other dense features is to concatenate the aspects with other dense features and pass them through an MLP (see Part (a) of Figure 3). In this case, one may measure the higher-order interaction of the aspects with other features. We denote this model by "Aug-Concat".

3.1.2. Wide and Deep Augmentation

We also test augmenting the aspects in a wide and deep format proposed by (Cheng et al., 2016) (see part (b) of Figure 3). In this case, the aspect embeddings are inputted to the wide part, and other dense features are inputted to the deep part of the architecture. This may alleviate the non-homogeneity issue of the concatenation embeddings from two different latent spaces of aspect embeddings and other features/embeddings. Also, using this architecture, one can measure the linear effect of each aspect dimension on the item's scores. We denote this model by "Aug-WD".

3.1.3. Tower Augmentation

Two-tower architecture (see Yang et al. 2020) is proposed to combine two different spaces like user and item embeddings, and generate a more unified concatenation of the embeddings. Inspired by user embedding augmentation, we also check the performance of this model when the aspect embeddings are input to one of the towers while other features are input to the other MLP tower (see part (C) of Figure 3). We denote this model by "Aug-2T".

3.1.4. Loss on Architectures

To be comprehensive, we also employed three different loss functions on top of each of the architectures. We use ListMLE (Chen et al., 2009), pairwise-cross-entropy (Boudiaf et al., 2020), and NDCG (Mohapatra et al., 2018) loss functions, when fitting the architectures. We add a suffix of "-LMLE", "-PCE", and "-NDCG" to denote the ListMLE, pairwise-cross-entropy, and NDCG loss used in each architecture.

4. Empirical Results

In this section, we investigate whether generating LLM-based aspects and using them in settings like anchor-based recommendation can help improve ranking metrics. Specially, we study the performance of Aug-Concat, Aug-WD, and Aug-2T combined with -MLE, -PCE, and -NDCG loss functions. We consider two benchmark models: (i) a regular Feed-Forward Neural Net (FFNN) with dense feature inputs, and (ii) FFNN with a Pre-training Language Model (PLM) embeddings (see Cer et al. 2018) concatenated with other dense features which we denote by PLM-FFNN. Like the augmentation architecture, we use three aforementioned loss functions on top of the embeddings generated by these benchmark architectures as well.

In our experimentation, we consider the concatenation of item description and item reviews to be the textual data and generate three aspects using PaLM2 for each item.

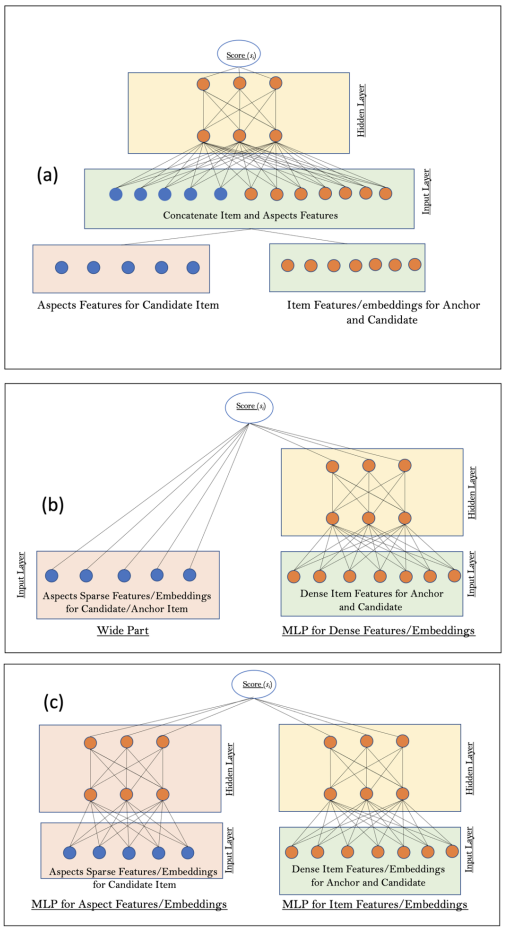

Figure 3. Augmentation Architectures

4.1. Data

We use a user-item interaction proprietary data set obtained from an eCommerce platform for our empirical validations. For the homogeneity of aspects generated by PaLM2, we restrict the items to be randomly selected only from the Toys&Games category.

Table 1. MRR and NDCG scores for the benchmark and LLM-based aspect augmented models

Model	NDCG@5	NDCG@10	MRR@5	MRR@10
FFNN-MLE	0.3750	0.4710	0.3229	0.3582
FFNN-PCE	0.3803	0.4775	0.3303	0.3649
FFNN-NDCG	0.3800	0.4768	0.3297	0.3645
PLM-FFNN-MLE	0.3727	0.4747	0.3261	0.3588
PLM-FFNN-PCE	0.3942	0.4918	0.3420	0.3775
PLM-FFNN-NDCG	0.3872	0.4861	0.3351	0.3710
Aug-Concat-MLE	0.4029	0.5044	0.3525	0.3868
Aug-Concat-PCE	**0.4280**	**0.5191**	**0.3700**	**0.4027**
Aug-Concat-NDCG	0.4259	0.5174	0.3690	0.4019
Aug-WD-MLE	0.4005	0.4976	0.3440	0.3790
Aug-WD-PCE	0.4151	0.5086	0.3571	0.3908
Aug-WD-NDCG	0.4169	0.5101	0.3591	0.3927
Aug-2T-MLE	0.4059	0.5017	0.3485	0.3830
Aug-2T-PCE	0.4179	0.5109	0.3598	0.3932
Aug-2T-NDCG	0.4170	0.5102	0.3592	0.3928

The data set includes 174k users and 139k item samples from user activity sessions from March 2023 to May 2023. The data include features like interaction history features (e.g. number of transactions in the last 30 days), general item features (e.g. item price, average rating), and textual features.

4.2. Aspects Generation

The overall number of uniquely generated aspects is 5,550 for about 139,000 items. Figure 4 shows the log frequency of generated aspects per item by our prompting approach. We observe that the generated aspects have the long-tail property. While some aspects are repeated for many items, most of the aspects are unique to a small number of items. About 80% of the aspects are assigned to less than ten items.

4.3. Results

The results are presented in Table 1. Comparing the performance between the FFNN model with augmented versions, we observe that all augmented models perform better than those only using the dense features. This justifies the effectiveness of applying feature augmentation techniques. The improvement of PLM augmented models is marginal compared with that using the LLM-based aspect augmentations. This indicates that implicit aspects, especially those from LLM, might provide more information and hence be important for building the ranking models. Across all augmented models, augmenting the LtR architectures with LLM-based aspect embeddings lead to the best performance in ranking metrics, which might be due to the larger number of parameters of this model. Note that in our architecture design, the hidden layers have the same dimensions within each MLP tower (for each MLP tower, we considered two layers with 64 and 8 neurons).

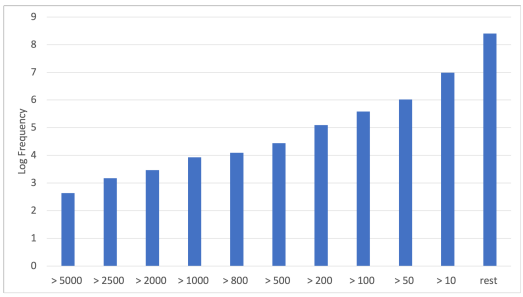

Figure 4. Log-Frequency of Aspects per Items. The number under each bar shows the number of aspects with at least that many items having those aspects.

5. Conclusion

Using LLMs has depicted promising results in various domains as they can potentially carry common-sense knowledge obtained from the body of the web's textual data. In this paper, we investigate the capability of LLMs in generating aspects for item recommendation to extract user intention keywords from user interaction data. We design a prompting approach to generate aspects from item descriptions and user reviews. Our experiments show that the generated aspects have long-tail property, and most aspects are unique to a small number of items. In our extensive experimentation, we use these aspects to generate embeddings and input them to ranking models of interest. More specifically, we test the performance of three augmentation approaches: (i) augmenting using feature concatenation, (ii) adding a wide aspect component beside a deep component of features, and (iii) aspect embedding tower augmentation.

Our experiments confirm that augmenting the ranking architectures using LLM-based aspects leads to an increase in relevancy metrics like MRR and NDCG, hence showing their capability to better describe the user choice behavior. We speculate that the reason for this improvement is due to using user intention keywords extracted from item description and user reviews. Also, our results indicate that augmenting using the feature concatenation approach attains higher relevancy scores when compared to other approaches.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Boudiaf, M., Rony, J., Ziko, I. M., Granger, E., Pedersoli, M., Piantanida, P., and Ayed, I. B. Metric learning: cross-entropy vs. pairwise losses. *arXiv preprint arXiv:2003.08983*, 2020.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

Chen, J., Ma, L., Li, X., Thakurdesai, N., Xu, J., Cho, J. H., Nag, K., Korpeoglu, E., Kumar, S., and Achan, K. Knowledge graph completion models are few-shot learners: An empirical study of relation labeling in e-commerce with llms. *arXiv preprint arXiv:2305.09858*, 2023.

Chen, W., Liu, T.-Y., Lan, Y., Ma, Z.-M., and Li, H. Ranking measures and loss functions in learning to rank. *Advances in Neural Information Processing Systems*, 22, 2009.

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 7–10, 2016.

Gao, Y., Sheng, T., Xiang, Y., Xiong, Y., Wang, H., and Zhang, J. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524*, 2023.

Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., and He, Q. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3549–3568, 2020.

Hou, Y., Mu, S., Zhao, W. X., Li, Y., Ding, B., and Wen, J.-R. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 585–593, 2022.

Hou, Y., He, Z., McAuley, J., and Zhao, W. X. Learning vector-quantized item representation for transferable sequential recommenders. In *Proceedings of the ACM Web Conference 2023*, pp. 1162–1171, 2023a.

Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., and Zhao, W. X. Large language models are zero-shot rankers for recommender systems. *arXiv preprint arXiv:2305.08845*, 2023b.

Maragheh, R. Y., Giahi, R., Xu, J., Morishetti, L., Vashishtha, S., Nag, K., Cho, J., Korpeoglu, E., Kumar, S., and Achan, K. Prospect-net: Top-k retrieval problem using prospect theory. In *2022 IEEE International Conference on Big Data (Big Data)*, pp. 3945–3951. IEEE, 2022.

Mohapatra, P., Rolinek, M., Jawahar, C., Kolmogorov, V., and Kumar, M. P. Efficient optimization for rank-based loss functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3693–3701, 2018.

Ni, J., Li, J., and McAuley, J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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)*, pp. 188–197, 2019.

Song, W., Wang, S., Wang, Y., and Wang, S. Next-item recommendations in short sessions. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pp. 282–291, 2021.

Wang, L. and Lim, E.-P. Zero-shot next-item recommendation using large pretrained language models. *arXiv preprint arXiv:2304.03153*, 2023.

Wang, W., Lin, X., Feng, F., He, X., and Chua, T.-S. Generative recommendation: Towards next-generation recommender paradigm. *arXiv preprint arXiv:2304.03516*, 2023.

Yan, A., Cheng, S., Kang, W.-C., Wan, M., and McAuley, J. Cosrec: 2d convolutional neural networks for sequential recommendation. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 2173–2176, 2019.

Yang, J., Yi, X., Zhiyuan Cheng, D., Hong, L., Li, Y., Xiaoming Wang, S., Xu, T., and Chi, E. H. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion Proceedings of the Web Conference 2020*, pp. 441–447, 2020.

Zhang, J., Xie, R., Hou, Y., Zhao, W. X., Lin, L., and Wen, J.-R. Recommendation as instruction following: A large language model empowered recommendation approach. *arXiv preprint arXiv:2305.07001*, 2023.