NEURAL RE-RANKING FOR PERSONALIZED HOME SEARCH

Group Members: Giacomo Bugli, Luigi Noto, Guilherme Albertini

Mentors: Shourabh Rawat, Niranjan Krishna, Andreas Rubin-Schwarz

INTRODUCTION

Search Systems aim at providing an ordered list of search results from an exhaustive set of documents that are most pertinent to a user. To do this, search systems are composed of multiple stages (see Figure 1), where following a search query from the user, a candidate set of items is generated and a ranking model (L_2 Layer) is applied to display the most relevant documents to the user.

However, the obtained rank may be sub-optimal due to an inability to model contextual factors (users' intents and mutual influence between items in the list). In practice, users often compare multiple items on a result page before generating a click action, meaning that information from other items in the same ranked list could affect a user's decision on the current item of interest.

This poses the need to include an additional component, the Re-ranking layer (L_3 Layer) which captures such relationships. Therefore, the focus of this project is in performing the re-ranking of an initially ranked list using the user's preference behaviour and items' cross interactions.

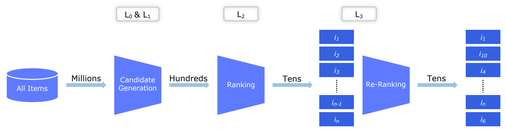

Figure 1. Stages of Search Systems.

DATASET AND METRICS

Zillow dataset:
Dataset owned by Zillow Group containing user search sessions over the length of a day. For each search session, we are provided with a list of items in the order they have been shown to the user as well as the binary user's interactions with such items (y_i^{click}, y_i^{submit}, $y_i^{favorite}$). Additionally, we have been provided with user-item interaction features ($\boldsymbol{pv}_i$), i.e. learned representation of user historic behavior, and item features ($\boldsymbol{x}_i$).

To evaluate the performances of the models produced during our work, we used the following metrics:

Normalized Discounted Cumulative Gain (NDCG@k):
The Discounted Cumulative Gain (DCG) is a measure of the usefulness, or gain, of a document based on its position in the ranked list. The gain is then discounted based on the rank of the item in the list. The DCG accumulated at a rank position k is then defined as

$$DCG@k = \sum_{i=1}^{k} \frac{rel_i}{\log(i+1)}$$

where rel_i is the relevance of item in position i of the list. In order to make this measure invariant to the length of the ranked list, the DCG is normalized by the Ideal Discounted Cumulative Gain (IDCG), which is the maximum possible DCG obtained by sorting the list of documents by their relevance and computing the DCG. The NDCG formula at a given position k is then given by

$$NDCG@k = \frac{DCG@k}{IDCG@k}$$

Mean Average Precision (MAP@k):
The Precision@k is defined as the fraction of relevant documents in the first k positions of the ranked list. Since Precision is invariant to the order of the items in the list, to weight the items based on their rank we define Average Precision (AP). The AP at a rank position k is then defined as

$$AP@k = \frac{\sum_{i=1}^{k} Precision@i \cdot rel_i}{\sum_{i=1}^{k} rel_i}$$

Then the MAP@k for a set of queries Q is the average of the AP@k over all queries.

$$MAP@k = \frac{\sum_{q=1}^{|Q|} AP_q@k}{|Q|}$$

METHODS

The main idea of re-ranking is to build the scoring function by encoding items cross-interactions into feature space. Many state-of-the-art methods for encoding the feature vectors are RNN-based, where the initial list is fed into the RNN-based structure sequentially and the encoded vector is output at each time step. The problem with these approaches is that they have limited ability to model the interactions between items in the list since the feature information of the previous encoded item degrades along with the encoding distance.

Then to model the mutual influences between items, we adopt:

- **PRM** Transformer-encoder model proposed by Pei et al. (2019), with a slight modification in the input layer (see Figure 2 for full model architecture)

The PRM model is trained by optimizing the **ListNet** (Cao et al., 2007) loss defined as

$$\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{pv}), \boldsymbol{y}) = -\sum_i \text{softmax}(\boldsymbol{y})_i \cdot \log(\text{softmax}(f_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{pv}))_i)$$

where $f_{\boldsymbol{\theta}}$ is the PRM scoring function, $\boldsymbol{x}$ is the input item features matrix, $\boldsymbol{pv}$ is the input user-item feature matrix, and $\boldsymbol{y}$ is the list of ground-truth relevance labels.

Selected Baselines:

- **No Re-rank** Initial ranked list obtained by Zillow preexisting ranking model
- **Popularity Baseline** Re-ranking based on the items click-through rate
- **LambdaMART** Traditional Learning-to-Rank (LTR) method proposed by Burges (2010), an ensemble model that is built on the MART (Multiple Additive Regression Trees) structure that utilizes gradient boosting, coupled with the concept of swap values called *lambdas*

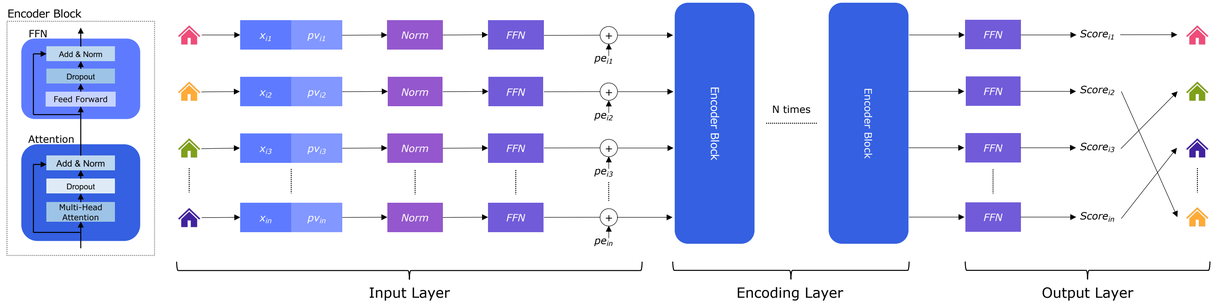

Figure 2. Detailed structure of PRM model and its sub-modules. *Norm* stands for Normalization layer, *FFN* for Feed Forward Network, and *pe*, for learnable positional encoding.

RESULTS

We trained PRM with two different targets:

- **PRM (Binary)** Here as binary ground-truth relevance we used the click interactions $rel_i = y_i^{click}$
- **PRM (Multi)** Here we used a multi-level ground-truth relevance constructed as follows

$$rel_i = y_i^{click} + 3y_i^{favorite} + 5y_i^{submit}$$

PRM shows improvements with respect to the selected baselines in both binary and multi-level relevance training settings. In particular, we have been able to obtain a lift of 2% on the NDCG@10 with respect to Zillow initial ranking model (No Re-rank case).

From Figure 5 and 6 we can clearly see how PRM re-ranks the items such that the relevant ones are pushed to the top.

Models	NDCG@5	NDCG@10	NDCG@15	NDCG@20	MAP@5	MAP@10	MAP@15	MAP@20
No Re-rank	34.18 %	39.82 %	43.24 %	45.35 %	27.79 %	30.57 %	32.08 %	32.92 %
Popularity	23.73 %	29.95 %	33.77 %	36.44 %	17.91 %	20.75 %	22.32 %	23.31 %
LambdaMART (Binary)	28.61 %	34.96 %	38.69 %	41.11 %	22.54 %	25.62 %	27.25 %	28.20 %
PRM (Binary)	**35.78** %	**41.79** %	**45.13** %	**47.21** %	**38.31** %	**38.22** %	**37.47** %	**36.89** %

Table 1. Ranking metrics computed on test set for the binary relevance based on items' clicks.

Models	NDCG@5	NDCG@10	NDCG@15	NDCG@20	MAP@5	MAP@10	MAP@15	MAP@20
No Re-rank	34.18 %	39.82 %	43.24 %	45.35 %	27.79 %	30.57 %	32.08 %	32.92 %
Popularity	23.73 %	29.95 %	33.77 %	36.44 %	17.91 %	20.75 %	22.32 %	23.31 %
LambdaMART (Multi)	28.40 %	34.79 %	38.52 %	40.96 %	22.33 %	25.43 %	27.06 %	28.02 %
PRM (Multi)	**35.61** %	**41.62** %	**44.96** %	**47.02** %	**38.11** %	**38.03** %	**37.30** %	**36.71** %

Table 2. Ranking metrics of models trained on multi-relevance targets but computed on the binary relevance based on items' clicks.

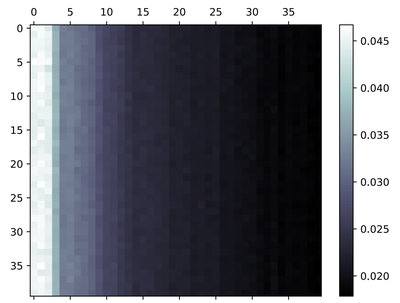

Figure 3. Average attention weights of 8 attention heads of the last transformer block for PRM (Binary).

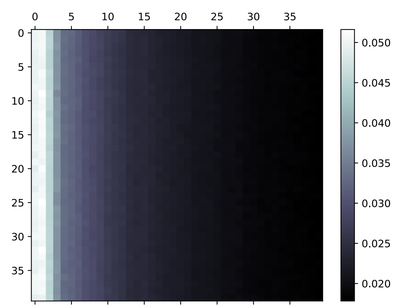

Figure 4. Average attention weights of 8 attention heads of the last transformer block for PRM (Multi).

CONCLUSIONS AND FUTURE WORK

Utilizing the current state-of-the-art for context-aware re-ranking we have been able to obtain an improvement in both NDCG and MAP metrics with respect to Zillow's generated ranking.
We have seen that by including additional information in the target used for training (multi-level relevance) we obtain slightly lower performances in both NDCG and MAP. The reason for this might be the increased sparsity of the labels or the fact that we used the best parameter configuration found in the PRM (Binary) cross-validation for PRM (Multi).

Future Directions:

- Explore more parameter configurations for PRM (Multi)
- Experiment training the PRM model without using the user-item interaction features to assess the impact of personalization
- Model evaluation on different types of interactions (e.g. favorite, submits) taken independently
- Instead of a single day of data, consider a broader time range to explore potential distribution shifts

REFERENCES AND ACKNOLWEDGEMENTS

Acknowledgements:
We would like to thank our mentors Niranjan Krishna, Andreas Rubin-Schwarz, Shourabh Rawat for their assistance throughout the project and their efforts in getting us the dataset to work with.

References:
Burges, C. J. C. (2010). From RankNet to LambdaRank to LambdaMART: An Overview (Techreport MSR-TR-2010-82). https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview/

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., & Li, H. (2007). Learning to Rank: From Pairwise Approach to Listwise Approach. Proceedings of the 24th International Conference on Machine Learning, 129–136. https://doi.org/10.1145/1273496.1273513

Pei, C., Zhang, Y., Zhang, Y., Sun, F., Lin, X., Sun, H., Wu, J., Jiang, P., Ge, J., Ou, W., & Pei, D. (2019). Personalized Re-Ranking for Recommendation. Proceedings of the 13th ACM Conference on Recommender Systems, 3–11. https://doi.org/10.1145/3298689.3347000

Pobrotyn, P., Bartczak, T., Synowiec, M., Bialobrzeski, R., & Bojar, J. (2020). Context-Aware Learning to Rank with Self-Attention. CoRR, abs/2005.10084. https://arxiv.org/abs/2005.10084

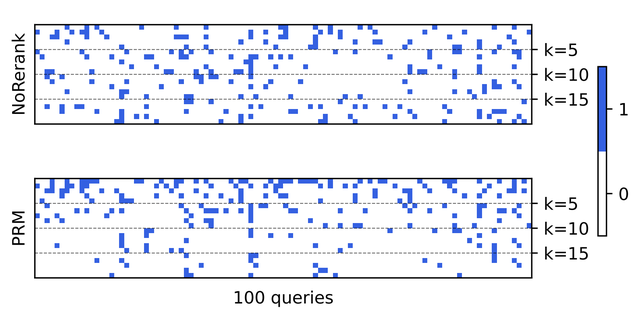

Figure 5. Visualization of PRM (Binary) re-ranking results against the No Re-rank baseline for the top-20 items on 100 randomly sampled test queries. The color encodes the item's ground-truth binary relevance based on clicks.

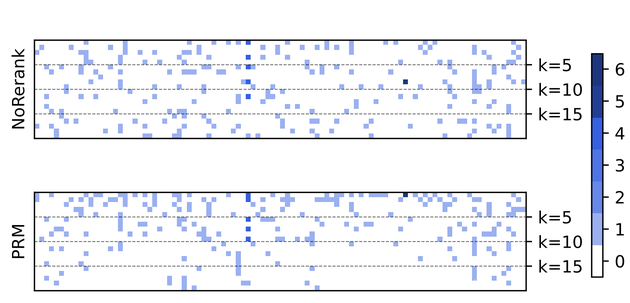

Figure 6. Visualization of PRM (Multi) re-ranking results against the No Re-rank baseline for the top-20 items on 100 randomly sampled test queries. The color encodes the item's ground-truth multi-level relevance.