Towards Efficiency and Sparsity: Exploiting Personalized Terms for News Recommendation

Anonymous Author(s)

1 INTRODUCTION

With the ever growing data and improving technologies, more and more news containing countless topics and events overwhelms people. It is nearly impossible for an individual to seek something she's interested in by herself. Luckily, online news platforms e.g. MSN[1] do that for each user, significantly alleviating the information overload problem [?]. The core function of such platforms is personlaized news recommendation.

News recommendation comprises of two major steps: recall and rerank: the former recalls a small subset of news that is relevant to the user's preference from the entire news collection; the latter reranks this subset in the descending order of click probability. In both stages, efficiency is the top concern [? ?]. As a result, two-tower structure prevails in news recommendation scenario. It represents each news as a compact vector in semantic space and aggregates historical news representations into a user profile in the same space, then takes the similarity between them as the click probability. Both encoding processes can be done offline, thus greatly speeds up model inferring.

Recently, large-scale pretrained language models (PLM in short) e.g. BERT [?] greatly outperforms shallow and light-weighted neural models in NLP field, which implies their potential to improve news recommendation accuracy. Under the two-tower setting, Wu et al. [?] first integrates PLMs to be the news encoder and validates their advantages in effectiveness. However, PLMs brings much more latency in training and inferring.

2 RELATED WORK

In this section, we review the related work of news recommendation, feature selection and candidate recall.

2.1 News Recommendation

News recommendation has been widely explored for decades. Traditional collaborative filtering methods [? ?] hash similar users into the same group by LSH before recommending. They either employ Matrix Factorization [?] to gather users. MF decomposes the user-item matrix to map users and items into the same latent space, where the inner product between their vectors captures the interaction score. Unlike movies or products, enoumous news is spawn every second, outstriping the increase of users, which makes the user-item matrix especially sparse. Factorization Machine [?] then introduces real-valued features to MF to alleviate the sparsity, but it requires manual features which is time and labor demanding.

Content-based news recommendation then emerges to address the above issues. Early works of content-based methods still rely on some manual features such as trend [?], geographics [?], and demographics [?] to model news and users. In the recent ten years, deep learning shows an unlimited potential and prevails in the task of representation learning [?]. So more and more works are underway to design equisite structures to learn representation of news and users directly from raw texts and browsing history respectively, taking the dot product between them as the click probability. This *two tower* workflow appeals to industry since encoding of news and user can be done offline, greatly improving model efficiency. Under the two tower setting, Wu et al. [? ? ? ? ? ? ?] proposes effective models that employs CNN, RNN, multi-head attention, and personalized attention to represent news and users. External information such as knowledge [?] and user-item biparititie graph [? ?] are also incorporated to enhance representations.

2.2 Efficient Transformers

More recently, large-scale pretrained language models (PLM in short) e.g. BERT [?] demonstrates impressive improvements over shallow and light-weighted neural models in NLP field. Though a previous work [?] integrates PLMs to news recommendation and validates their improvements, it is yet non-trivial to efficiently implement them: the quadratic complexity of self-attention w.r.t. the input sequence length poses an intense challenge to speed up encoding users with dozens of historical articles that may contain thousands of words in total. Researchers make a lot of efforts to lessen this problem. SpeedyFeed [?] deduplicates user's historical news and candidate news in a batch, significantly speeding up the training process. Apart from optimizing training scheme in industry, more research modifies the self-attention layer to reduce complexity. For example, Longfomer [?] sparsifies the full self-attention to a sliding and dialated pattern, which reduces the complexity to linear w.r.t. the input length; Fastformer achieves the same result by additive attention and element-wise production. It also reaches a new state-of-the-art performance on MIND [?], a large-scale dataset in news recommendation.

Another line of research followes a feature selection intuition that prunes the input before expensive interaction. Hofstätter et al. [?] restricts BERT to only inspect top K important passages per document. It splits the selection and ranking stage, where the former is trained in teacher-student paradiagm by the pseudo labels produced by a BERT, and the latter only scores the selected passages. Using the similar cascading setting, ?] extracts fewer valueble items from user history to feed into final ranking. However,

[1] https://www.msn.com/en-us/news

Conference'17, July 2017, Washington, DC, USA
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/10.1145/1122445.1122456

Type	**Methods**	**AUC**	**MRR**	**NDCG@5**	**NDCG@10**
Sota	FastFormer	**72.68**	**37.45**	**41.51**	**46.84**
Baselines	Two Tower	71.43	36.16	39.67	45.29
	TES-First	68.00	32.91	36.44	42.86
	TES-BM25	68.00	32.91	36.44	42.86
	TES-Entity	68.00	32.91	36.44	42.86
Ours	TES	69.62	34.30	37.47	43.21

cascading achitechture requires labels for each stage, which is prohibitive in our scenario because there is no ground-truth indicating terms that the user really favors. Gallagher et al. [?] explores framework for jointly optimizing cascade search, but it is not practical in a BERT-style model since it relies on specified empirical risk rather than ranking loss. The first application of sparse selection in news recommendation is SFI [?]. It sparsely and automatically selects important historical news before effective candidate-aware interaction, guaranteeing the efficiency and effectiveness of the model. Despite its improvements, SFI executes selection at new level, possibly bringing high bias to the final ranking. It also neglects PLMs to promote performance due to its one tower limitation.

In our work, we select the browsing history at word-level to keep more fine-grained and comprehensive information instead of pruning the historical news set. With only a handful of personlaized terms, we apply PLMs to fully capture the interaction within and among historical articles, promoting the effectiveness of the news recommender to a new level at competitive speed.

2.3 Sparse Recall

As a bonus, personalized terms generated by our model can transform the entire user history to a rather short query, with which we can align the candidate recall to ad-hoc retrieval. Traditional retrieval techniques organize documents by inverted-index, and sort them by their frequency-based scores such as BM25 [?] and query likelihood [?] given a query. However, they face two main problems: non-contexts: frequency-based scores ignore contexts; semantic-mismatch: only the documents with exact match could be retrieved, where out-of-vocabulary harms the retrieval performance; numerous works are proposed to address them. DeepCT [?] and HDCT [?] compute contextualized scores instead of BM25. COIL [?] and SNRM [?] learn semantic-aware representation for each word, while keeping the sparsity to combine inverted index. Some works [? ? ?] expand documents and quries for wider reception field.

3 EXPERIMENT

3.1 Results