Unveiling Challenging Cases in Text-based Recommender Systems

Ghazaleh Haratinezhad Torbati[1], Anna Tigunova[1] and Gerhard Weikum[1]

[1]*Max Planck Institute for Informatics, Saarbrücken, Germany*

Abstract

In this paper we challenge the standard ways of how text-based recommender systems are trained and evaluated. We highlight the necessity of focusing on long-tail users and items, as those are the cases where the text-based prediction can potentially win over collaborative filtering methods. We also raise concerns of choosing datasets and data preparation for recommender training and evaluation. Finally, we reconsider the issue of how recommenders are evaluated, and propose drilling-down into different groups of users and items, as well as search-based evaluation as an alternative to solely measuring a global metric for context-free test points.

Keywords

Recommender Systems, User Reviews, Evaluation Modes

1. Introduction

Recommender systems is a mature field with a wide variety of powerful methods, based on matrix factorization, auto-encoders, transformers and other machine learning models. The models fall into the categories of interaction-based, content-based and hybrid approaches. The content-based paradigm includes methods to leverage textual user reviews when available. This paper focuses on such text-centric recommenders.

There are numerous experimental studies for evaluating the performance of recommender systems. As real application data is in the realm of big companies, most studies are based on benchmarks with excerpts of various datasets, split into train, validation and test folds. Apart from some critical studies (e.g., [1, 2]), most benchmark-based works report strong performance, giving the impression that there is not much room for improvement (for the basic setting known as "direct recommendation", as opposed to conversational, cross-domain etc.). This paper reconsiders the state of the art in experimental evaluation, points out shortcomings and proposes amendments to obtain more realistic insight on the difficulty and quality of predicting items for users.

Specifically, we question and tackle the following concerns:

- **Choice of dataset:** With the limited availability of public datasets, the specific choice is crucial for the achievable performance.

Perspectives on the Evaluation of Recommender Systems Workshop (PERSPECTIVES 2023), September 19th, 2023, co-located with the 17th ACM Conference on Recommender Systems, Singapore, Singapore.

ghazaleh@mpi-inf.mpg.de (G. H. Torbati); tigunova@mpi-inf.mpg.de (A. Tigunova); weikum@mpi-inf.mpg.de (G. Weikum)

CEUR Workshop Proceedings (CEUR-WS.org)

- **Data preparation:** Many experiments remove long-tail or otherwise data-poor instances, thus *avoiding difficult cases.* Some experiments *include easy cases* where training and test instances are semantically related, for example, recommending songs by an artist that has been seen at training (for the same user), or books written by the same author.
- **Choice of user text:** As the entirety of a user's review text may be too large to consume in a single training batch, there are often assumptions on which specific parts are selected.
- **Global metrics:** Virtually all literature solely reports on aggregate metrics (i.e., Precision, NDCG, MRR or AUC) over the entire test set. This misses out on obtaining insight into how well recommenders work for *specific groups of items and users.*
- **Training samples:** Some settings have only (or mostly) positively labeled data (as users rarely bother marking negative cases). Many works then sample negative training instances uniformly at random from the pool of all unlabeled data. Potentially, this is a biased situation where positive and negative samples are easier to discriminate.
- **Test candidates:** Typically, evaluations are based on "context-free" predictions per user, that is, no context other than the user's training points and a set of withheld test points with one or more (known to be) positive and a larger set of randomly picked negative instances. This does not consider the more realistic case where users need *situative recommendations*, based on fine-grained search queries or a specifically liked item as context.

In the following, we focus on the issues of 1) dataset choice and data preparation, 2) choice and processing of user text, 3) refining experimental results by interesting subgroups of users and items, and 4) selecting test candidates by simulating the situative context of user search.

We make the full experimental data and hyperparameters publicly available on https://personalization.mpi-inf.mpg.de/PERSPECTIVES23

2. Dimensions of Experimental Evaluations

2.1. Choice of Domain and Data

Existing recommendation datasets, enriched with textual review data, cover different domains, including movies[1], restaurants[2] and consumer products[3]. Not all domains are suitable for review-based recommendation task, though: for instance, the reviews on purchased products often reflect the condition of the product, packaging or delivery, which does not tell anything about the product features that are appealing to the user. On the other hand, reviews describing films or music mostly concern their content. Yet, they also frequently suffer from low quality, because user engagement into watching a film or listening to a song is minimal - thus yielding shallow and sometimes copied texts.

In this paper we run a benchmark on two datasets from book communities. Books have more long-tailed niche items, as opposed to movies or songs, where most users are interested in popular items (e.g., blockbuster movies, top hits in music charts), giving rise to a heavily

[1] https://grouplens.org/datasets/movielens

[2] https://www.yelp.com/dataset

[3] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2

Table 1
Statistics of different datasets with user reviews. AM stands for Amazon, GR for Goodreads.

	#items	#users	#reviews	#u per item	#i per user	avg review len
AM Digital Music	456,979	840,329	1.6M	3.47	1.88	42
AM Beauty	32,585	324,023	371.3k	11.39	1.14	44
AM Toys and Games	624,786	4.205k	8.2M	13.13	1.95	41
MovieLens 1M	3,416	6,040	—	292.7	165.5	—
Yelp 2015	60,785	366,715	1.6M	25.8	4.3	126
GR-10K-dense	385,660	10,000	1.4M	3.74	144.16	144
GR-10K-random	271855	10,000	679.4k	2.5	67.94	136
GR-10K-sparse	158,554	10,000	262.5k	1.66	26.25	126
AM-10K-dense	203,930	10,000	511.9k	2.51	51.19	190
AM-10K-random	92174	10,000	164.2k	1.78	16.42	108
AM-10K-sparse	58,443	10,000	79.3k	1.36	7.93	83

skewed distribution of items being likable. Specifically, we work with data compiled in the UCSD repository of recommender datasets https://cseweb.ucsd.edu/~jmcauley/datasets.html, gathered from Amazon reviews and the Goodreads online community. The Amazon (AM) data has been used occasionally in the literature (much less, though, than other slices of AM products such as electronics or toys), whereas the GR data has hardly been considered in prior research. Importantly, we use special care in preparing different samples of this very large data, with judicious control of sparseness and other levels of difficulty.

Table 1 shows statistics on various datasets, contrasting popular benchmarks with different data slices that we derive from the AM and GR crawls. Observe that the density of the number of users per item is one or two orders of magnitude lower for the books data.

The texts of book reviews are more involved than the ones given for music, toys or restaurants, the reason being that users who write reviews already spent many hours with the item, whereas movies, songs or restaurants are short-term episodes. This makes book reviews particularly difficult to process because they can be long, discuss multiple topics, contain contradictory sentiments and plenty of side information that is useless for prediction. For example, a review text that says “I am a retired teacher. I read many books. With this one, I spent a whole weekend in our summer cottage without any breaks ...” tells something about the user but does not give tangible cues about why the user liked this particular book.

Another challenge for book recommendations is the absence of negative data, because negative reviews with low ratings are very rare. This makes it impossible to cast the task as regression for rating prediction.

2.2. Data Preparation

Long-tail Users and Items: Commercial recommender systems aim to satisfy the vast majority of customers, who tend to have strong overlap in their preferences and often like mainstream items. In contrast, providing recommendations to discover new and less popular items for users with diverse interests remains largely unsolved. Similarly, rare items pose a challenging case. Approaches for zero-shot transfer (e.g., [3, 4]) typically require learning item-item similarities from dense data, which is not easily available for the book domain.

Several studies explicitly cut off long-tail users and items, for example, by keeping only users whose number of reviews is above a certain threshold (e.g, [5, 6]), typically in the order of 10. Additionally, these studies select only the top 90% of users and items, based on sorting by lengths and numbers of reviews. Even works on handling cold-start users make such assumptions and prefer popular, but easy, benchmarks like MovieLens with ample "blockbusters" that almost everybody likes (e.g., [7]).

In our experiments, we alleviate these restrictions to allow for data-poor users and rare items. We impose the only restriction for the users to have at least 3 reviews, so that it is possible to split them into training/validation/test sets. As for items, We also include books that are completely unseen during training, as we aim to model realistic applications with new items appearing at a high rate.

Controlling Data Sparseness: Interaction-based recommender systems, like collaborative filtering and auto-encoders, are extremely widespread and stand behind the majority of commercial applications. The success of these methods, however, relies on the connectivity of the interaction graph between users and items. When the graph is sparse, the interaction model does not have enough data for informed inference. Leveraging text reviews is a way to approach the problem of a sparse graph. Yet, both hybrid and text-only models still face a data sparseness issue.

Our proposal for more refined evaluation obtains insights into the performance on different slices of the interaction graph, including areas of low density and of high density. These subsets are sampled in a controlled way for stress-testing methods in a spectrum of operational regimes. More details are given in Section 3.1.

Eliminating Duplicate and Highly Similar Items: Some benchmark data contains near-duplicate items, like different editions of the same book or versions of the same song (e.g., in some datasets of the UCSD respository [8], not always removed by the researchers that use this data). When versions of the same item exist in both training and test sets, the outcome includes trivial predictions. This simple observation is sometimes overlooked in previous studies.

Our data preparation ensures that the items from the same group (e.g., books written by the same author, or songs by the same band) are, for each user, disjoint in train and test/eval splits. Overlap between different users is acceptable, though.

2.3. Choice of User Text

Ideally, a user's text-based profile is comprised of all reviews ever posted, and the recommender would automatically learn which sentences and phrases are the most informative cues. However, this can easily result in very long texts, which are difficult and expensive to process, especially with state-of-the-art Transformer models. BERT, for example, has a 512-token limit for its input. Even when the model allows much longer inputs (e.g., GPT-3 and GPT-4), there is a computational, monetary and energy-consumption cost associated with each additional token.

To overcome this problem, several studies simply crop the input text to the desired length [9, 10]. This approach may work well with encyclopedic texts, where the main message is often contained in the first couple of sentences. However, in book reviews, useful hints on user preferences are spread across the entire texts. Moreover, with a very tight token budget, the text ingestion may even have to consider only a subset of reviews, and it is not clear how to meaningfully sort the reviews so that the cropping ingests the best ones.

Another type of strategy is splitting the input into chunks or selecting sentences and aggregating the resulting representations or scores (e.g, by max-pooling) [11, 12]. This is a popular approach as it allows information flow from the entire text. However, the different chunks do not interact with each other inside the model; so the method is still bound to be heuristic and far from optimal.

Some studies restrict their content choice only to the titles or the genres/categories and other tags of a user's books (e.g., [13, 14]). Item titles are rarely descriptive of the contents, though, and genres of items may be too broad (e.g., only 10 coarse-grained genres in the Goodreads data at the UCSD repository).

A better approach is to selectively pick highly informative portions of a user's texts to build a concise profile, either from sentences or phrases. Reviews often include personal facts, emotional expressions or very specific statements, which do not support creating a concise image of the user's content preferences. In this study we propose and investigate different approaches for judiciously selecting sentences or n-grams, to create compact profiles that fit into the model's allowed token budget. We compare these to established means of capturing content features, like item titles and category/genre tags.

2.4. Global Metrics

Typically, experimental evaluations or recommender systems report global performance numbers averaged over the whole dataset. However, different points or regions in the data exhibit varying degrees of difficulty for the model. By averaging over all data points, the results are dominated by the common cases, which are easiest to learn. The performance of the model on long-tail instances is neglected.

For book recommender systems, the long tail of difficult cases is formed by users with different reading habits: some are *sporadic* readers, having read very few books, but possibly from different genres; other users are *bibliophilic*, where the challenge is to come up with good suggestions to someone who has already read a lot. Similar considerations apply to items: some books are *totally new* or fall into the *niche* of books that attract little attention, whereas others are more *mainstream.*

Taking care of these corner cases is important for user satisfaction where "no user is left behind" [15], and the full diversity of items is exploited. This refinement resembles established techniques for breaking the data down into head and tail items or users. However, there is no prior work that looks into the refined combinations of groups. With these considerations in mind, we experiment with subgroups of users and items as explained in Section 4.3.

2.5. Test Candidates

In practical settings it is not feasible to provide a user with a complete ranking of all items, so as to suggest the next book to read. In addition, such a global ranking may not even suit the user's needs, as the "next book to read" may depend on the user's situative context. A standard way out, in lab tests, is to solely provide the predicted rank of a withheld positive item with respect to a randomly sampled subset of negative items (for books, drawn randomly from unlabeled items, as there are hardly any truly negative labels). This approach, however, tolerates many "easy" negatives: the ones that are obviously wrong for the user (e.g., a *children book* as a negative for someone who reads only *sci-fi*). Moreover, with different sampling techniques, albeit all random but with different distributions, performance results become incomparable across recommenders. Evaluating on the entire dataset would eliminate this bias, but still faces the uncertainty as to whether an unlabeled item would be positively perceived by a user. The only way to overcome this problem would be to carry out online user studies, but this is beyond scope of most academic labs, at least at sufficiently large scale (cf. [16]).

In our study, we cannot solve this problem either, but aim to shed more light into its nature, by showing that the selection of negative samples for evaluation has a substantial influence. We propose an alternative way of performance evaluation that better resembles real use cases. We do this by simulating a search-based request where a user starts with a query or a particularly liked book and asks for recommendations of similar books [17, 18]. Instead of uniformly sampling from all unlabeled items, the negative samples are obtained with respect to the user's query, by retrieving a set of approximate matches to the positive item, say top-100 based on scores from IR models such as BM25 ranking. The query is derived from the positive item, by taking its title, categories and descriptions as textual input for IR model. This brings negative test points much closer to the positive test item, and makes the predicted ranking much harder (see [19, 20] for similar approaches). The positive query item or seed item is added to the candidate set, and the final ranking of this pool is computed by the actual recommender system.

2.6. Other Concerns

Selection of negative training samples is vital for learning a robust model. In the setup where almost all item labels are either positive or unknown, the typical strategy is to randomly pick unlabeled items to serve as negatives. In reality, however, it is unclear whether these items may be liked by the user if the user had a chance to see them. To properly tackle this issue would require large user studies, which are typically beyond the capabilities of academic research, or as mitigation, better strategies for negative sampling (e.g., [21, 22, 23, 24, 25, 26]).

3. Experimental Setup

3.1. Data

As discussed in Section 2.1, we use two book datasets, based on the UCSD repository [8]:

- **Goodreads:** a sample of the Goodreads online community[4], with ~10M interactions between ~1.5M items and ~280K users, together with item information (title, description and genres), ratings and users reviews.
- **Amazon-Books:** a sample of Amazon reviews for books with ~29M interactions between ~2.3M items and ~3.1M users, including item information (title, description and categories) and user ratings and reviews.

[4] https://www.goodreads.com/

In this work we consider only positive interactions: the ones which are associated with ratings ≥ 4, as ratings below 3 and truly negative reviews are extremely rare.

3.2. Models

We benchmark performance using a suite of configurations for a state-of-the-art **BERT-based two-tower transformer** architecture. We prefer transformers over alternatives such as CNNs [27], as it has shown superior behavior in prior works and comes with a pre-trained language model. Specifically, BERT [28] is trained to learn representations of user and item texts, restricted to 128 tokens each. While such a tight budget may seem overly restrictive on first glance, we keep in mind that there is a computational and environmental cost for each token and we have to expect larger texts for full-scale user data from real applications.

On top of BERT, we use a feed-forward network for the learned predictions. The entire model is fine-tuned end-to-end using binary cross entropy for predicting whether the given user likes the given book. We run experiments on NVIDIA Quadro RTX 8000 GPUs with 48 GB memory and implement all models via PyTorch.

3.3. Evaluation metrics

We evaluate by two metrics:

- **NDCG@5:** normalized discounted cumulative gain for the top-5 recomendations. This metric is most relevant when users merely inspect a handful of top results, without having to scroll.
- **Precision@1:** fraction of correct predictions for top-1. This metric is most relevant for smartphones where users typically look at only one recommendation.

Other metrics, like MRR, AUROC and NDCG@k for other cut-offs k, were observed to behave nearly proportionally to NDCG@5; hence omitted here. Unlike prior works, we do not just report global numbers but drill-down on specific groups of user-item combinations.

4. Findings

4.1. Choice of Dense vs. Sparse Data

We evaluate performance for three choices of data samples from AM and GR: *random*, *sparse* and *dense*. These slices are obtained as follows.

To ensure that the resulting graphs are connected, we perform two steps of random user and item selection, followed by a third step that is specific to the desired data characteristics. First, we pick 500 users and sample 2000 books of the selected users, both uniformly at random. This gives us a pool of users who have common interest with the initial 500 users.

- **Random Data.** We pick 10K users (minus the initial 500) uniformly at random. We take the complete set of books liked by the sampled users.
- **Dense Data.** This proceeds analogously to the random case, but the users are sampled with probabilities *proportional to their cumulative item degrees.*

- **Sparse Data.** This proceeds analogously to the random case, but the users are sampled with probabilities *inversely proportional* to their cumulative item degrees.

We provide statistics for all variants in Table 1.

Table 2
Micro-averaged NDCG@5 for different data choices.

	Amazon			Goodreads		
Method	dense	random	sparse	dense	random	sparse
CF	32.13	20.07	12.05	47.35	35.85	18.98
genres	45.81	32.49	25.8	51.46	46.02	35.27
titles$_{rand}$	46.79	35.88	27.57	**54.02**	**49.46**	42.75
reviews$_{rand}$	**47.05**	**37.36**	**29.91**	53.72	49.29	**43.96**

The results for the variants of both datasets are shown in Table 2 for four configurations:

- **CF:** a standard collaborative filtering method based on the Funk matrix factorization [29], as a reference point.
- **genres:** the two-tower transformer using only genre tags for all books of a user.
- **titles**$_{rand}$**:** the same architecture, using book titles. If a user has many books so that the total length of all concatenated titles exceeds the input limit of the transformer, a subset of titles is selected uniformly at random.
- **reviews**$_{rand}$**:** the same architecture, with a random selection of sentences from all reviews of a user, up to the allowed input size.

From Table 2 we observe:

- NDCG@5 (and other metrics as well, not shown here) degrade as we move from dense to sparse data, for both AM and GR.
- CF is viable only for dense and random slices, and outperformed by the text-based methods even on dense data, likely because our dense variant is not as highly connected as other benchmarks in the literature, such as MovieLens.
- All text-based methods perform similarly, with titles and review sentences being best.

Overall, this shows that the chosen degree of data sparseness has a massive influence.

4.2. Choice of User Text

We compare the following choices for 128-token text input:

- **genres**: genre tags of the user's books.
- **titles**$_{rand}$: randomly selected book titles, up to the token budget.
- **reviews**$_{rand}$: randomly selected sentences from all reviews of the user.
- **reviews**$_{idf}$: top sentences ranked by idf scores, up to the token budget.
- **reviews**$_{sbert}$: top sentences from reviews ranked by Sentence-BERT similarity to any sentence of the item description, up to the token budget.
- **reviews**$_{3gram}$: top 3-grams from reviews ranked by tf-idf scores, up to the token budget.

As the selection of text matters most when the interaction graph is sparse, we focus on results for the 10K-sparse slices of AM and GR for the rest of the paper. Similar trends, with weaker amplitude, are observed for the dense and random slices, too. The results are given in Table 3.

Table 3
Micro-averaged NDCG@5 and P@1 on 10K-sparse Data for different input text selection choices.

	Amazon		Goodreads	
Method	NDCG@5	P@1	NDCG@5	P@1
genres	25.8	14.66	35.27	20.21
titles$_{rand}$	27.57	16.4	42.75	26.01
reviews$_{rand}$	29.91	17.47	43.96	26.83
reviews$_{idf}$	**31.09**	**18.22**	43.79	26.91
reviews$_{sbert}$	30.97	18.19	43.86	26.6
reviews$_{3gram}$	29.52	16.73	**44.2**	**27.17**

We make the following salient observations:

- Leveraging reviews gives a significant edge over the simpler techniques with genres or titles.
- Between the reviews-based techniques, the idf-scored sentence selection performs best on AM, and the 3-gram technique is best on GR. This indicates that judicious text selection matters, but does not require a lot of sophistication.

4.3. Drill-down into User and Item Groups

To obtain deeper insight into the recommender performance for different groups of users and items, we split data points as follows.

Items: We split test items into **seen** and **unseen** points, depending on whether they were present (for any user) during model training or appear only for evaluation. This is a crude proxy for distinguishing mainstream vs. long-tail items. Given the limited size of our data, a more fine-grained break-down does not make sense here.

Users: We split the set of users, by their numbers of reviews:

- **Sporadic (spo)** users are the 50% with the least numbers of reviews.
- **Regular (reg)** users are the ones that lie between 50% and 90% in the interaction (i.e., review-count) distribution.
- **Bibliophilic (bib)** users are those 10% with the highest numbers in the review-count statistics.

To determine the threshold for the user groups, we examined the distribution of the number of books per user. The distribution function has major shifts, with sharp increases, at the 50% quantile and the 90% quantile: we call the 50% lowest users *sporadic* and the ones above the 90% quantile *bibliophilic*; the remaining ones in the middle are referred to as *regular* users.
The performance numbers for the resulting 6 groups of item-user combinations are shown in Tables 4 for AM and 5 for GR.

Table 4
Micro-averaged NDCG@5 grouped by users and items on AM-10K-sparse.

Method	spo-unseen	reg-unseen	bib-unseen	spo-seen	reg-seen	bib-seen
CF	0.0	0.0	0.0	**47.58**	38.48	28.54
genres	16.68	21.88	33.26	23.83	28.6	32.84
titles$_{rand}$	17.88	24.93	36.6	22.34	28.74	33.5
reviews$_{rand}$	17.82	24.15	34.62	34.39	38.48	**41.24**
reviews$_{idf}$	18.7	25.39	**36.81**	34.65	**39.32**	41.07
reviews$_{sbert}$	**20.08**	**25.61**	36.8	33.4	37.31	40.8
reviews$_{3gram}$	17.89	24.12	35.29	32.59	36.28	39.76

Table 5
Micro-averaged NDCG@5 grouped by users and items on GR-10K-sparse.

Method	spo-unseen	reg-unseen	bib-unseen	spo-seen	reg-seen	bib-seen
CF	0.0	0.0	0.0	**55.64**	44.28	34.0
genres	13.38	22.39	34.68	37.29	43.8	45.0
titles$_{rand}$	25.88	**35.63**	41.59	43.12	49.92	48.73
reviews$_{rand}$	23.5	34.46	40.55	52.11	**54.1**	50.75
reviews$_{idf}$	24.71	34.07	39.8	53.42	53.62	**51.05**
reviews$_{sbert}$	**26.0**	35.52	**41.62**	51.46	52.35	48.7
reviews$_{3gram}$	24.73	34.6	41.33	53.37	53.81	49.96

The results suggest the following key findings:

- By design, CF fails on unseen items. However, it performs well on seen items for sporadic users. One conjecture for this behavior is that these users have a fairly narrow taste: their books are centered on a single genre and topic theme (e.g., *vampire romance*, which is frequent in these datasets). As long as CF has a few training points, it can make good predictions for these low-diversity users.
- In the case of unseen items, the simple technique of selecting random titles works amazingly well (not necessarily the best but very competitive).
- The smartness of the more judicious text selection methods pays off in the regime of seen items for regular and bibliophilic users. In these cases, the recommender can learn from longer texts by users who wrote more reviews. It is notable, though, that the performance numbers for bibliophilic users are lower than for regular users. Our educated guess (based on inspecting instances) for explaining this phenomenon is that the users with many books also exhibit highly diverse taste: their books are spread across a wider range of genres and content topics. So they pose a more demanding case for recommender systems.

Overall, these findings tell us that it is important to look into specific user and item groups, in order to obtain a deeper assessment of how well (or not so well) recommenders perform.

4.4. Standard versus Search-based Evaluation

All previous experiments were performed with standard evaluation where negative test points are drawn uniformly from all unlabeled data (100 negative test points for each positive test point).

This setup promotes getting unrealistically high results, as many negatives are easy cases to dismiss by the trained recommender. Therefore, we studied the alternative mode of **search-based** evaluation, where negative test points are based on textual similarity to the positive point. Specifically, we use the BM25 retrieval technique over the text representations of the items to obtain the top-100 related but negative test points, starting with the positive point as a seed query. This mimics the situative context of a user who asks for similar books after having enjoyed reading the positive item. With top-100 closest negative items retrieved, we add the positive item and run the evaluation on the set of 101 test points.

Table 6
Micro-averaged NDCG@5 of different evaluation modes for 10K-sparse data.

	Amazon		Goodreads	
Method	Standard	Search-based	Standard	Search-based
CF	12.05	9.71	18.98	13.86
genres	25.8	6.47	35.27	12.71
titles$_{rand}$	27.57	8.38	42.75	14.38
reviews$_{rand}$	29.91	**9.93**	43.96	15.75
reviews$_{idf}$	**31.09**	9.41	43.79	15.58
reviews$_{sbert}$	30.97	9.44	43.86	15.69
reviews$_{3gram}$	29.52	8.61	**44.2**	**16.05**

Table 6 shows the results for search-based evaluation, side-by-side with standard evaluation. The main findings from this experiment are the following:

- The NDCG@5 numbers are dramatically lower for the search-based evaluation, compared to the standard mode. This underlines our hypothesis that the literature overstates high numbers, and tends to disregard the much harder but more realistic case of search-based recommendation.
- The smart text-selection methods are still superior to simpler techniques, but their gains become smaller in search-based mode. Moreover, CF performs almost on par with the other methods (but also loses big compared to standard evaluation).
- All methods heavily struggle with the complexity of discriminating the one positive test item from the textually similar negative items. In fact, this reflects a more general problem: in reality, these are just unlabeled points; they are treated as negatives but the user may actually like some of them if she were to be asked. Without performing large-scale user studies, there is no way to resolve this issue, though.

5. Related Work

Several studies noted that the evaluation setups in lab settings are mostly easier than in real life. For instance, Zhang et al. [2] criticize randomly created evaluation splits, as different parts of the dataset may vary in difficulty. Sachdeva and McAuley [30] discuss the drawbacks of pruning long tail from the dataset.

Some authors discuss the challenges of dealing with review textual input. Lin et al. [31] highlight the importance of thoroughly designing input text selection, to overcome input length limitation; Wu et al. [32] claim that the common way of representing the items and users via item titles is inefficient.

Taking special attention to different user and item groups is described in [33], where authors emphasize the importance of novel and comprehensive recommendations for the users with unusual tastes. Thus, their study, as well as the one of Li et al. [15], investigate model's performance on different user groups (clusters). At the same time, Li et al. [15] warn that the predictions of the transformer-based models might be biased towards popular items, due to their prevalence in the model's pretraining data.

Specific support for long-tail items and users has been addressed in the literature mostly under the theme of cold-start and zero-shot recommendations (e.g., [3, 34]). A typical approach is to embed cold-item features into the same space as warm items, thus learning relatedness among items. This assumes that cold items come with tags and descriptions. For the user side, that assumption is much less practical: users would not likely expose a rich profile when they are new to a community or merely occasional contributors. In this data-poor regime, the best option is to leverage whatever few reviews a user has provided. Such textual cues are rarely considered in the cold-start and zero-shot literature.

In the absence of negative labels, most works (including ours), draw negative training samples from a static distribution. Alternative approaches include mixes of uniform and popularity-based sampling Yang et al. [23], or techniques for bias correction during training Yi et al. [22]. Haratinezhad Torbati et al. [26] explored sampling with respect to the item genres and weighting points by similarity to the user's profile, following techniques for PU learning [24]. Zhang et al. [21] and Chen et al. [25] propose methods for adaptive sampling, iteratively over training epochs. For the books domain with virtually no negative labels, this issue is still not satisfactorily solved.

6. Conclusion

This work discusses previously underexplored issues in the evaluation of text-based recommender systems. We reviewed the importance of carefully controlling the data sparseness, as our experiments showed that many models can lose up to 3x times performance on extremely sparse data. Additionally, we examined different ways of selecting the textual input for models with a tight limit of input tokens, showing that judicious techniques for selection significantly improve performance over simply sampling random sentences. A detailed drill-down into different groups of users and items shows that global aggregate metrics fail to reveal valuable insights into specific strengths and weaknesses of methods. Finally, we studied a search-based evaluation mode, which is more realistic and results in substantial drops in quality for most recommender models.

References

[1] M. Ferrari Dacrema, P. Cremonesi, D. Jannach, Are we really making much progress? a worrying analysis of recent neural recommendation approaches, in: RecSys '19, ACL, New York, NY, USA, 2019, p. 101–109.

[2] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Comput. Surv. 52 (2019).

[3] J. Li, M. Jing, K. Lu, L. Zhu, Y. Yang, Z. Huang, From zero-shot learning to cold-start recommendation, in: AAAI '19, 2019, pp. 4189–4196.

[4] R. Raziperchikolaei, G. Liang, Y. Chung, Shared neural item representations for completely cold start problem, in: RecSys '21, ACM, 2021, pp. 422–431.

[5] C. Chen, M. Zhang, Y. Liu, S. Ma, Neural attentional rating regression with review-level explanations, in: WWW '18', 2018, pp. 1583–1592.

[6] H. Liu, Y. Wang, Q. Peng, F. Wu, L. Gan, L. Pan, P. Jiao, Hybrid neural recommendation with joint deep representation learning of ratings and reviews, Neurocomputing 374 (2020) 77–85.

[7] P. Li, R. Chen, Q. Liu, J. Xu, B. Zheng, Transform cold-start users into warm via fused behaviors in large-scale recommendation, in: SIGIR '22, ACM, 2022, pp. 2013–2017.

[8] J. McAuley, Recommender Systems and Personalization Datasets, 2022. URL: https://cseweb.ucsd.edu/~jmcauley/datasets.html.

[9] B. Xiao, X. Xie, C. Yang, Y. Wang, Rtn-gnnr: Fusing review text features and node features for graph neural network recommendation, IEEE Access 10 (2022) 114165–114177.

[10] T. Wang, Y. Fu, Item-based collaborative filtering with BERT, in: ECNLP '20', ACL, Seattle, WA, USA, 2020, pp. 54–58.

[11] R. A. Pugoy, H.-Y. Kao, Unsupervised extractive summarization-based representations for accurate and explainable collaborative filtering, in: ACL-IJCNLP '21, ACL, Online, 2021, pp. 2981–2990.

[12] R. A. Pugoy, H.-Y. Kao, BERT-based neural collaborative filtering and fixed-length contiguous tokens explanation, in: AACL '20, ACL, Suzhou, China, 2020, pp. 143–153.

[13] Q. Zhang, J. Li, Q. Jia, C. Wang, J. Zhu, Z. Wang, X. He, Unbert: User-news matching bert for news recommendation, in: Z.-H. Zhou (Ed.), IJCAI '21, 2021, pp. 3356–3362.

[14] C. Wu, F. Wu, S. Ge, T. Qi, Y. Huang, X. Xie, Neural news recommendation with multi-head self-attention, in: EMNLP-IJCNLP '19, ACL, Hong Kong, China, 2019, pp. 6389–6394.

[15] R. Z. Li, J. Urbano, A. Hanjalic, Leave no user behind: Towards improving the utility of recommender systems for non-mainstream users, in: WSDM '21, ACM, New York, NY, USA, 2021, p. 103–111.

[16] P. Castells, A. Moffat, Offline recommender system evaluation: Challenges and new directions, AI Magazine 43 (2022) 225–238.

[17] Q. Ai, Y. Zhang, K. Bi, X. Chen, W. B. Croft, Learning a hierarchical embedding model for personalized product search, in: SIGIR '17, ACL, New York, NY, USA, 2017, p. 645–654.

[18] G. H. Torbati, A. Yates, G. Weikum, You get what you chat: Using conversations to personalize search-based recommendations, in: ECIR '21', 2021, pp. 207–223.

[19] G. H. Torbati, A. Yates, G. Weikum, Personalized entity search by sparse and scrutable user profiles, in: CHIIR '20', ACL, New York, NY, USA, 2020, p. 427–431.

[20] J. Liu, Z. Dou, G. Tang, S. Xu, Jdsearch: A personalized product search dataset with real queries and full interactions, in: SIGIR '23, ACL, New York, NY, USA, 2023, p. 2945–2952.

[21] W. Zhang, T. Chen, J. Wang, Y. Yu, Optimizing top-n collaborative filtering via dynamic negative item sampling, in: SIGIR '13, ACM, 2013.

[22] X. Yi, J. Yang, L. Hong, D. Z. Cheng, L. Heldt, A. Kumthekar, Z. Zhao, L. Wei, E. Chi, Sampling-bias-corrected neural modeling for large corpus item recommendations, in: RecSys '19, ACL, New York, NY, USA, 2019, p. 269–277.

[23] J. Yang, X. Yi, D. Zhiyuan Cheng, L. Hong, Y. Li, S. Xiaoming Wang, T. Xu, E. H. Chi, Mixed negative sampling for learning two-tower neural networks in recommendations, in: WWW '20, ACL, New York, NY, USA, 2020, p. 441–447.

[24] J. Bekker, J. Davis, Learning from positive and unlabeled data: a survey, Machine Learning 109 (2020).

[25] J. Chen, D. Lian, B. Jin, K. Zheng, E. Chen, Learning recommenders for implicit feedback with importance resampling, in: WWW '22, ACM, 2022.

[26] G. Haratinezhad Torbati, G. Weikum, A. Yates, Search-based recommendation: The case for difficult predictions, in: WWW '23, ACL, New York, NY, USA, 2023, p. 318–321.

[27] L. Zheng, V. Noroozi, P. S. Yu, Joint deep modeling of users and items using reviews for recommendation, in: WSDM '17, ACM, New York, NY, USA, 2017, p. 425–434.

[28] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[29] S. Funk, Netflix Update: Try This at Home, 2006. URL: https://sifter.org/~simon/journal/20061211.html.

[30] N. Sachdeva, J. McAuley, How useful are reviews for recommendation? a critical review and potential improvements, in: SIGIR '20, ACM, New York, NY, USA, 2020, p. 1845–1848.

[31] J. Lin, X. Dai, Y. Xi, W. Liu, B. Chen, X. Li, C. Zhu, H. Guo, Y. Yu, R. Tang, et al., How can recommender systems benefit from large language models: A survey, arXiv preprint arXiv:2306.05817 (2023).

[32] L. Wu, Z. Zheng, Z. Qiu, H. Wang, H. Gu, T. Shen, C. Qin, C. Zhu, H. Zhu, Q. Liu, et al., A survey on large language models for recommendation, arXiv preprint arXiv:2305.19860 (2023).

[33] J. Šafařík, V. Vančura, P. Kordík, Repsys: Framework for interactive evaluation of recommender systems, in: RecSys '22, ACM, New York, NY, USA, 2022, p. 636–639.

[34] B. Liu, B. Bai, W. Xie, Y. Guo, H. Chen, Task-optimized user clustering based on mobile app usage for cold-start recommendations, in: KDD '22, ACM, 2022, pp. 3347–3356.