Advanced Information Retrieval

There Are No Games

Mihály Barosi
Rachel Hold
Orhidea Shatri
Moritz Ziegner

Group 25

January 13, 2026

1 Introduction

Digital distribution platforms for games, such as Steam, offer a large amount of games. This makes it difficult for many games to become visible to players, especially for indie developers who have limited marketing resources. Meanwhile popular games from Large Studios and Publishers often attract an extensive amount of players and benefit from the success of earlier titles, which in turn helps later releases become easy to discover. Often times players prefer games that share similar themes, gameplay or story elements with games that they already like. Therefore, recommending games based on similarity to popular game titles has been proposed as a way to improve discoverability [1].

The video game market is growing quickly and it includes a lot of new releases, indie games and classic games, with many games released every month. With these numerous choices of available games out there, users can find it difficult to discover games that match their interests. Recommendation systems can benefit players by suggesting new games in a personalized way [2]. In this project we analyse the following research question: Does using a fine-tuned SPLADEV3 model improve the findability of games compared to the default SPLADEV3 model and a BM25 baseline?

2 Related Work

Many text search systems are based on BM25, which is a ranking function that comes from the Probabilistic Relevance Framework [3]. In this framework, relevance is treated as something that cannot be directly observed and documents are ranked by how likely they are to be relevant to the information need of the user.

Recent work looks at improving lexical search to learn which words from the vocabulary are important for each query and document, while still using inverted indices. SPLADE follows this idea by adding a log-saturation effect to the term weights. This helps prevent single words from becoming too strong and keeps the representations sparse. In the experiments, SPLADE reaches performance that is comparable to strong dense retrieval models and keeps the number of operations close to standard methods that use bag of words representations. Because of this, SPLADE can be used as a first-stage retrieval model that combines sparse and efficient retrieval with competitive ranking performance [4].

Another approach to retrieval makes use of dense vector representations instead of relying on matching exact words. Dense Passage Retrieval maps each question and passage to vectors and ranks passages based on the dot product between those vectors. The training data for the model contains a question, one relevant text passage and several irrelevant text passages, including random passages and passages retrieved by BM25. In the evaluation, the dense retriever outperforms the BM25 baseline regarding top-k retrieval accuracy on most of the datasets [5].

Previous work has focused on game recommender systems that use behavioural data from players on Steam. The recommendations rely on implicit player data such as which games users own and how long they have been playing them [6]. These approaches mainly focus on behavioural recommendation with implicit feedback rather than on retrieval using game descriptions.

3 Experiments and Results

This section presents the experiments and results. The GitHub repository can be found here: repository [7].

3.1 Experiments

We used OpenSearch [8] as our BM25 baseline (as OpenSearch uses BM25 internally). Additionally we used a default SPLADEV3 model [9] to also compare the performance differences of standard SPLADEV3 to our custom fine-tuned model. Our dataset was taken from the Steam Dataset 2025: Multi-Modal Gaming Analytics Dataset [10] from kaggle.com, from which we used the files `applications.csv` and `reviews.csv`. For simpler usability we used the sentence-transformers [11] pip package.

3.1.1 Data Preprocessing

We carried out this step in the files `csv_shortener.py` and stored the results in the file `steam_games_cleaned.csv`. In addition, to save on resources during the training process, we decided to only include applications in the games category. For the training and retrieval steps we only needed the columns `name`, `short_description` and `appid` from the `applications.csv` file [10]. For reading the data from the csv files we used the pandas library [12]. After this process our dataset was left with 142.629 entries.

3.1.2 BM25 - Baseline

As BM25 [3] is widely used as a baseline in information retrieval research, we also decided to use BM25 in our work as a comparison against both the normal and our fine-tuned SPLADE model. As OpenSearch uses the BM25 ranking mechanism internally [13], we decided to use it as our BM25 implementation. To accomplish this, we first loaded our dataset into OpenSearch utilizing default settings in `index_data.py`. After that point we used the built in search method in our file in `bm25_search.py`, to return the answers of our query which are ranked inside of OpenSearch by BM25.

3.1.3 SPLADEV3 - Second Baseline

As we fine-tuned and used a SPLADE model [9] we decided on also using a standard version of the same model in comparison to our model. We ran the encode locally in `splade_base_encode.py`. For reproducibility we provide access to our generated encodings in the readme of our GitHub repository. The retrieval itself was done in the `splade_base_retrieval.py` file, where it was also ran locally.

3.1.4 Fine-tuned SPLADEV3

As was the case above we ran and fine-tuned our SPLADEV3 Model [14] locally as well. The fine-tuning was done with the help of the SparseEncoderTrainer from the sentence transformers library [11], with hyper-parameters, which can be found in `fine_tuner.py`, taken from a Hugging Face blogpost about fine-tuning sparse encoder models [15]. We tokenized the game's description, adding its title to its token pool, and trained the model to retrieve the right title based on a random set of tokens from this set. We used a "SparseMultipleNegativesRankingLoss" loss function [16] due to it being made as a loss function with negatives training, which is important as every entry in the dataset represents its own game, so the model needs to easily recognize different entries. We used a 40k/5k/5k train/validation/test split from our cleaned dataset of our 142k entries, to mitigate potential issues with over-fitting. The encoding and retrieval steps were performed as mentioned in the previous SPLADEV3 section.

3.2 Results

In this section we go over the results of the experiments, the means of evaluation, and what they mean for our research question.

3.2.1 Machine Evaluation

Our Machine Evaluation test set consists of the 5k test split mentioned in Fine-tuned SPLADEV3. The automatic evaluation was carried out in the `steam_games_evaluation.py` file. The evaluation was carried out with SparseSparseInformation-RetrievalEvaluator from the sentence transformers library using default settings and a batch_size of 16. See 5.2 for all metrics.

3.2.2 Manual Evaluation

In addition to automatic evaluation, we also manually evaluated the base SPLADEV3 model, the fine-tuned model, and the BM25 model. In total, 63 queries were evaluated, with each query returning 5 game titles most relevant to the query. Each returned title was graded with a score: 2 = Highly relevant, 1 = Somewhat relevant, and 0 = Not relevant.

Some queries were simple genre names, e.g., "sci-fi" while others considered multiple components, e.g., "Zombie fps" and some referenced other game titles, e.g., "Games like Terraria". Relevancy was determined by checking the game's description, tags, about section, and images, and comparing it against the query. If all query components could be associated with the title, it was graded a 2. If some, but not all components could be associated, the title was graded a 1. If the title was entirely unrelated, it was graded a 0.

For example, for the query "Creature collection game like pokemon", the following are some of the titles returned by the fine-tuned SPLADE model:

- Soul Locus. This is a "Creature Collection" game and that "It's Pokémon meets tower defence" [17]. It fully matches the query and was graded a 2.
- Whisper Forest. This title features some creature collection elements, but is not similar to Pokémon, therefore, it was graded a 1 [18].
- Collector. Despite the description calling this a "Collector game", it is not a **Creature Collector**, which is a particular game genre [19]. It also makes no references to Pokémon and was graded 0.

The fine-tuned SPLADE model in particular commonly returned the titles "game" and "Game Name" for queries including the word "game". Both of these titles are clear empty or template listings not intended to be discoverable or recommended to users, so they were left ungraded and omitted for all further calculations [20, 21].

After grading the returned titles, the query failure rate (how often a query returned at least one irrelevant title), precision, nDCG, MAP and MRR were calculated using `calc_eval_metrics_manual.py`. See results in 2.

3.3 Interpretation

Based on the machine driven evaluation in 5.2, the fine-tuned SPLADEV3 model appears to perform better than the base SPLADEV3 model. However, this directly contradicts the findings of the manual evaluation. According to the manually checked average relevance grade 5,6,4, base SPLADEV3 and BM25 return similarly relevant game titles, while the fine-tuned SPLADE model returns less relevant titles, but with higher variance.

Based on the calculated metrics in 2, the BM25 model had the lowest failure rate, i.e., the lowest incidence of not-relevant returned titles, followed by base SPLADEV3, with the highest failure rate belonging to the fine-tuned SPLADE model.

We hypothesise that the fine-tuned SPLADE model was more likely to return titles with shorter descriptions, especially those which contained tags that were also present in the query. This behaviour may have been encouraged by the fine-tuning phase, and not caught by the machine evaluation, which was implemented using similarly structured test queries.

The fine-tuning method outlined in Fine-tuned SPLADEV3 may have caused a bias towards games shorter descriptions, and games whose titles and descriptions contained tags included in the query. In the manual evaluation query set, there were four queries which included the word "simulator"/"simulation". Out of the 20 titles returned for these queries, 16 were in one of the following titles: "Kissing Simulator"; "Emo Simulator"; "Protest Simulator"; "Piano Simulator"; "FootballLife"; "bridg". A commonality between these titles is that they all have short descriptions containing either "simulator" or "simulation".

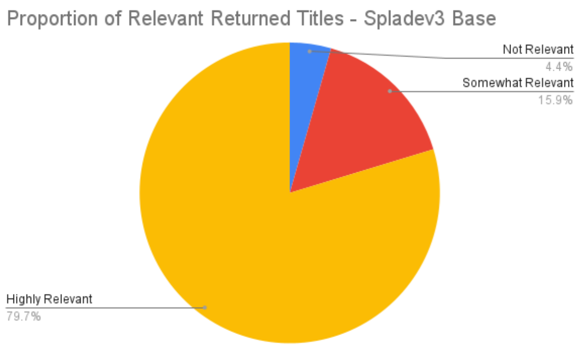
4 Conclusion

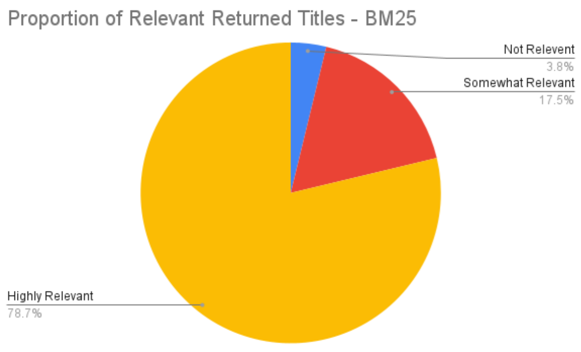
In this project we assessed whether using a fine-tuned SPLADEV3 model could improve the findability of games using the default SPLADEV3 model and BM25 as a baseline. The machine evaluation gave us strong results for our fine-tuned model, but the manual evaluation showed the opposite. For the manual judgments, the fine-tuned model produced on average less relevant results and had the highest failure rate, whereas the default SPLADEV3 and BM25 returned more relevant results for the queries. Overall the default SPLADEV3 performed the best, followed closely by BM25 and our fine-tuned model performed the worst. Therefore, our results do not support the hypothesis that our fine-tuned SPLADEV3 improves the findability of games.

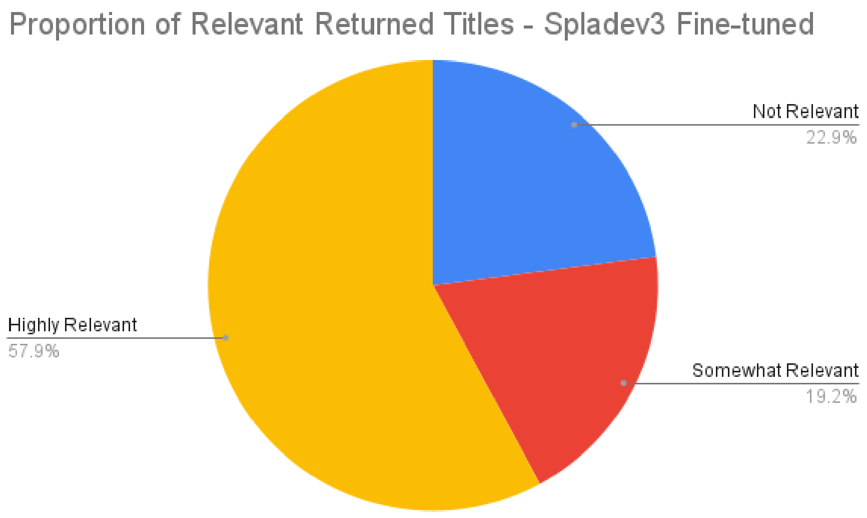
We hypothesis that fine-tuning on our dataset did not improve general search quality and may have pushed the model to unwanted patterns in the data: appearing to show bias towards games with shorter, more basic descriptions – games which users are typically less interested in. Future work would focus on improving the better aligning the fine-tuning and machine evaluation phases with typical human usage. Filtering could also improve performance, for example, by removing placeholder, incomplete, or low effort game entries.

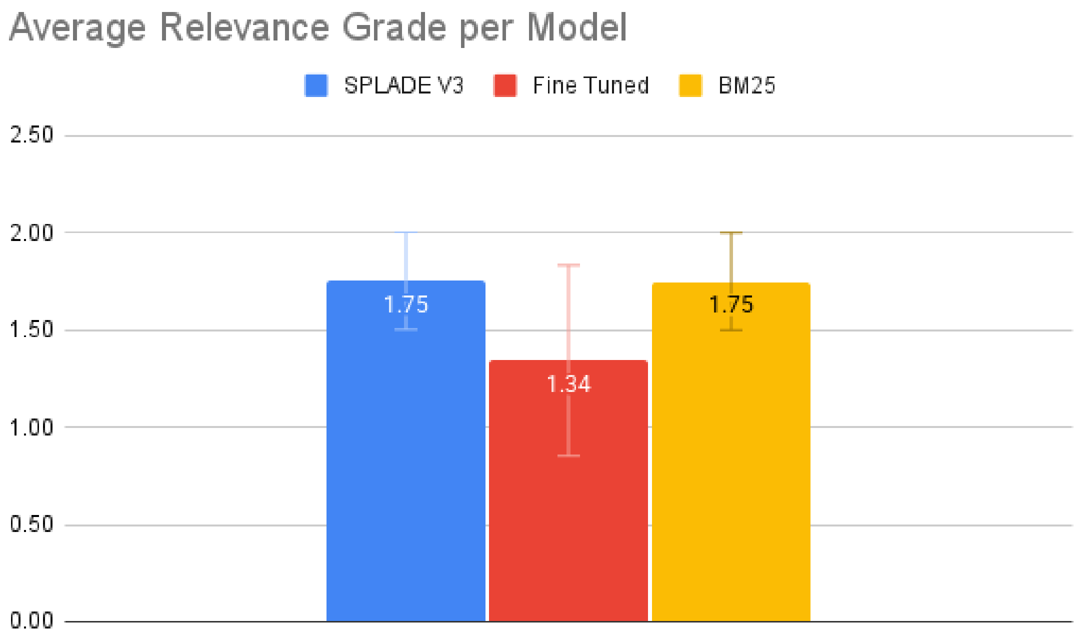
5 Appendix

5.1 Relevance Metrics Visualised









5.2 Machine Evaluation

Metric	Default SPLADEV3	Fine-tuned SPLADEV3
accuracy@1	0.7785	0.8917
accuracy@3	0.8525	0.9592
accuracy@5	0.8832	0.9748
accuracy@10	0.9130	0.9888
precision@1	0.7785	0.8917
precision@3	0.2842	0.3197
precision@5	0.1766	0.1950
precision@10	0.0913	0.0989
recall@1	0.7785	0.8917
recall@3	0.8525	0.9592
recall@5	0.8832	0.9748
recall@10	0.9130	0.9888
ndcg@10	0.8445	0.9435
mrr@10	0.8227	0.9286
map@100	0.8257	0.9292
query_active_dims	27.4402	107.8732
query_sparsity_ratio	0.9991	0.9965
corpus_active_dims	191.8756	98.3774
corpus_sparsity_ratio	0.9937	0.9968
avg_flops	4.8365	1.6232

Table 1: Machine evaluation results for Default and Fine-tuned SPLADEV3 models

5.3 Manual Evaluation

Metric	SPLADEV3	Fine-tuned SPLADEV3	BM25
Failure rate	0.143	0.476 (judged only)	0.111
Precision@5 / Precision	0.956	0.765 (judged only)	0.962
nDCG@5	0.965	0.822 (judged prefix)	0.958
MAP	0.980	0.832 (judged prefix)	0.981
MRR	0.992	0.829 (judged prefix)	0.984

Table 2: Manual evaluation results for Default SPLADEV3, Fine-tuned SPLADEV3 and BM25 models

Relevance	**SPLADEV3**	%	**Fine-tuned SPLADEV3**	%	**BM25**	%
0	0	0.00%	4	26.76%	0	0.00%
1	6	46.15%	4	26.76%	3	20.00%
2	7	53.85%	7	46.76%	8	80.00%

Table 3: Relevance Frequencies of Non-English Game Titles

	Per-Query Mean	**Per-Query Variance**
Mean	1.75	0.18
Variance	0.14	0.09

Table 4: Mean and Variance Metrics: Base SPLADEV3

	Per-Query Mean	**Per-Query Variance**
Mean	1.34	0.42
Variance	0.38	0.20

Table 5: Mean and Variance Metrics: Fine-tuned SPLADEV3

	Per-Query Mean	**Per-Query Variance**
Mean	1.75	0.16
Variance	0.14	0.07

Table 6: Mean and Variance Metrics: BM25

References

[1] N. Q. Vu and C.-P. Bezemer, “Improving the discoverability of indie games by leveraging their similarity to top-selling games: Identifying important requirements of a recommender system,” in *Proceedings of the 16th International Conference on the Foundations of Digital Games*, FDG ’21, (New York, NY, USA), Association for Computing Machinery, 2021.

[2] R. Bunga, F. Batista, and R. Ribeiro, “From implicit preferences to ratings: Video games recommendation based on collaborative filtering,” in *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, SCITEPRESS – Science and Technology Publications, Lda, 2021.

[3] S. Robertson and H. Zaragoza, “The probabilistic relevance framework: Bm25 and beyond,” *Foundations and Trends in Information Retrieval*, vol. 3, pp. 333–389, 01 2009.

[4] T. Formal, B. Piwowarski, and S. Clinchant, “SPLADE: sparse lexical and expansion model for first stage ranking,” *CoRR*, vol. abs/2107.05720, 2021.

[5] V. Karpukhin, B. Oguz, S. Min, L. Wu, S. Edunov, D. Chen, and W. Yih, “Dense passage retrieval for open-domain question answering,” *CoRR*, vol. abs/2004.04906, 2020.

[6] R. Sifa, C. Bauckhage, and A. Drachen, “Archetypal game recommender systems,” *CEUR Workshop Proceedings*, vol. 1226, pp. 45–56, 01 2014.

[7] “GitHub - Spivonxe/There-are-no-games: AI ML powered game recommender for AIR Group25 — github.com.” https://github.com/Spivonxe/There-are-no-games. [Accessed 12-01-2026].

[8] “OpenSearch — opensearch.org.” https://opensearch.org/. [Accessed 06-01-2026].

[9] C. Lassance, H. Déjean, T. Formal, and S. Clinchant, “Splade-v3: New baselines for splade,” 2024.

[10] D. Fountain, “Steam Dataset 2025: Multi-Modal Gaming Analytics — kaggle.com.” https://www.kaggle.com/datasets/crainbramp/steam-dataset-2025-multi-modal-gaming-analytics. [Accessed 06-01-2026].

[11] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 11 2019.

[12] T. pandas development team, “pandas-dev/pandas: Pandas,” Feb. 2020.

[13] “Learning to rank for amazon opensearch service.” https://docs.aws.amazon.com/opensearch-service/latest/developerguide/learning-to-rank.html#:~:text=OpenSearch%20uses%20a%20probabilistic%20ranking,25%20to%20calculate%20relevance%20scores. [Accessed 11-01-2026].

[14] M. Ziegner, “There-are-no-games (revision 2770988),” 2026.

[15] "Training and finetuning sparse embedding models with sentence transformers v5." https://huggingface.co/blog/train-sparse-encoder, july 2025. [Accessed 11-01-2026].

[16] M. Henderson, R. Al-Rfou, B. Strope, Y. hsuan Sung, L. Lukacs, R. Guo, S. Kumar, B. Miklos, and R. Kurzweil, "Efficient natural language response suggestion for smart reply," 2017.

[17] "Soul Locus on Steam — store.steampowered.com." https://store.steampowered.com/app/349190/Soul_Locus/. [Accessed 13-01-2026].

[18] "Whisper Forest on Steam — store.steampowered.com." https://store.steampowered.com/app/3010250/Whisper_Forest/. [Accessed 13-01-2026].

[19] "Collector on Steam — store.steampowered.com." https://store.steampowered.com/app/1172850/Collector/. [Accessed 13-01-2026].

[20] "game on Steam — store.steampowered.com." https://store.steampowered.com/app/1372870/game/. [Accessed 13-01-2026].

[21] "Game Name on Steam — store.steampowered.com." https://store.steampowered.com/app/2087640/Game_Name/. [Accessed 13-01-2026].