

When Do Graphs Beat Text? Knowledge-Graph Embeddings as a Content-Free Candidate-Generation Substrate for Recommendation

Karima Boutalbi
Cegedim Business Services
Paris, France
karima.boutalbi@cegedim.com

Olivier Le Van
Cegedim Business Services
Lyon, France
olivier.levan@cegedim.com

Rim Hantach
Engie
Paris, France
rim.hantach@external.engie.com

Rafika Boutalbi
Aix-Marseille Université, LIS Lab
Marseille, France
rafika.boutalbi@lis-lab.fr

Abstract

Retrieval-augmented and generative recommenders open with a nearest-neighbour candidate-generation step over frozen item embeddings, usually a text encoder over item descriptions. We study *that step in isolation*, no scoring model, no downstream generator, and ask whether a content-free knowledge-graph embedding over the item-attribute triples a catalogue already stores is competitive with text, across six catalogues spanning three themes (music, movies, books). The answer is not “graphs beat text”: under best-effort representations it is roughly even, and the direction is set by *complementarity*, not graph density, the KG wins only when its relations encode what the text lacks (social tags on KGRec-Music, where a content-free embedding beats even a 0.6B text encoder, 0.51 vs. 0.35), and text wins when the graph merely restates the item string. A cheap a-priori redundancy *heuristic* predicts the direction better than density but, frozen on four catalogues, calls Movies correctly and *misses* Books out of sample, so we frame it as a hypothesis, not a validated predictor at $n = 6$. Scope matters: against a behavioural collaborative filter, content only competes in the cold-start and long-tail regime, which is where the text-vs-KG choice lives. The substrate is *content-free*, a no-training SVD over the triples matches a tuned RotatE, though a symmetric text baseline is just as cheap. The result is an honest map of *when*, and with which encoder, content helps candidate generation.

CCS Concepts

• **Information systems** → **Recommender systems**; *Retrieval models and ranking*.

Keywords

knowledge-graph embeddings, candidate generation, retrieval-augmented recommendation, representation geometry, graph auto-encoders

ACM Reference Format:

Karima Boutalbi, Rim Hantach, Olivier Le Van, and Rafika Boutalbi. 2026. When Do Graphs Beat Text? Knowledge-Graph Embeddings as a Content-Free Candidate-Generation Substrate for Recommendation. In *Proceedings of GRAIL 2026: Workshop on Generative, Retrieval-augmented, and Agentic Intelligence for Personalization (GRAIL @ CIKM '26)*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 Introduction

Retrieval-augmented and generative recommenders all open with the same step. A set of candidates is selected by approximate nearest-neighbour search [11] over frozen item embeddings, and only then is that set reranked, explained or handed to a language model [7, 9, 14]. The generative stage operates on the candidates it receives, so the ceiling of the whole system is fixed by what the embedding retrieves. The default embedding is text: run a pretrained sentence encoder over each item’s title and description and index the result. This is convenient, but it inherits whatever the encoder keys on, and for many catalogues that is surface wording, not what makes two items substitutable.

Two cases from our datasets make the point (Figure 1). Bananarama’s *Cruel Summer* ships a plot-summary blurb (“alone in a hot city...”) but a knowledge graph of *social tags*, 1980s, dance-pop, summer, the blurb never states; its KG neighbours are other 1980s dance-pop songs by *different* artists (Madonna, Corey Hart), while its text neighbours, keyed on *hot/summer/alone*, run from Deftones to Kid Rock, musically unrelated. The advantage is not universal: Rammstein’s *Mein Herz Brennt* has a graph of only artist and genre, both already in its “Rammstein – ...” string, so KG and text retrieve the *same* neighbourhood, other tracks by the same artist, and the graph adds nothing. One distinction runs through both: the graph helps when *complementary* to the text (Bananarama), not when *redundant* with it (Rammstein), making that precise, and cheap to check in advance, is what this paper is about.

This paper asks when the knowledge-graph embedding is the better candidate-generation substrate, descriptively, on six catalogues across three themes (music, movies, books), for the retrieval step alone (no scoring model, no downstream generator). The graph is attractive because it reads no item text and needs no large pre-trained encoder, only the item-attribute triples a catalogue already

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
GRAIL @ CIKM '26, Rome, Italy
© 2026 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

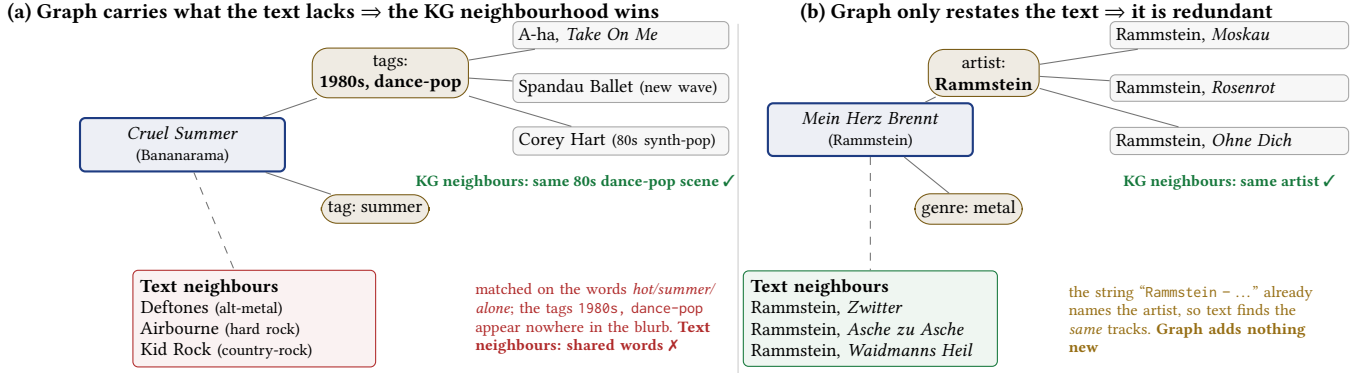


Figure 1: When does the knowledge graph help? Rounded blue boxes are the query item; ochre boxes are its graph attributes; grey boxes are real nearest neighbours retrieved by each encoder (KGRec-Music and MSD-A embeddings). (a) Bananarama's *Cruel Summer* has a graph of *social tags* (1980s, dance-pop, summer) that its plot-summary text never states. The KG-embedding neighbours share that era and scene, other 1980s dance-pop by *different* artists (✓), while the text-embedding neighbours are assembled from the words *hot/summer/alone* and span alt-metal to country-rock (✗). Here the graph is *complementary* to the text and the KG wins. (b) Rammstein's *Mein Herz Brennt* has a graph of only artist and genre, both already written into the "Rammstein - ..." text string. Both encoders return the same neighbourhood of same-artist tracks, so the graph is *redundant* and text alone suffices. Both cases are illustrative, hand-picked to show the mechanism, not a statistical sample; whether to index the graph turns on which regime a catalogue is in, which we make precise in Section 6.

stores; the question is whether that content-free representation can match a sentence encoder for retrieval.

On six catalogues we compare the two head-to-head, isolating the retrieval step from any scoring model. Both are geometrically well-conditioned, neither collapses onto a shared direction, so the comparison is about *what* each encodes, not about one being broken. A first look suggests a clean one-number rule: on the *shipped* embeddings of the four music catalogues that carry one, the sign of the KG-over-text advantage tracks the density of the graph (Pearson $r = 0.98$ at $K=20$), dense graphs favouring the KG. But that rule does not survive a fair fight. Even with a tuned graph embedding, the density correlation over all six catalogues is already looser ($r = 0.87$, Section 6); and once each item's *full best-effort* text replaces the shipped title, the density ordering breaks outright: a dense graph whose two relations (artist, genre) merely restate the text loses, while a sparser graph carrying composer and lyricist wins. The predictor that holds up is *complementarity*: the KG wins as a candidate generator exactly when it encodes attributes the text does not. Density is a useful but imperfect proxy for that, a fast first screen that a dense-yet-redundant graph defeats. Along the way we find the graph substrate can be even cheaper than assumed: a no-training TF-IDF+SVD over the raw triples matches a tuned RotatE on most datasets, at a fraction of the cost.

Contributions.

- A head-to-head *retrievability* comparison of text and knowledge-graph item embeddings for candidate generation, with best-effort representations on both sides, on six catalogues across three themes, isolating the retrieval step from any scoring model.
- The central finding: the KG wins exactly when its graph encodes item attributes the text lacks (*complementarity*).

Triples-per-item is a cheap first screen for this, but an imperfect proxy, a dense-yet-redundant graph (artist and genre already in the text) defeats it, so we show where the density rule holds and where it breaks, including on two non-music catalogues (movies, books; Amazon-KG v2.0) held out from the analysis.

- A cheap, content-free KG encoder: a no-training TF-IDF+SVD over the raw triples that matches or beats a tuned RotatE on three of the six datasets and a 0.6B text encoder on two. (Its symmetric no-training text baseline is just as cheap, so the practical contest is cheap-vs-cheap.)
- A retrieval-augmented-generation reading: the generator inherits whatever ceiling the pool sets, and over 1,756 users the retrieved text and graph contexts are complementary, conditioning on both beats either alone (tight bootstrap CIs), though fusing the two *retrieval* indices only helps by enlarging the candidate pool, not at a matched budget.

We also propose, but frame only as a hypothesis (Section 8), a cheap redundancy score, computable before indexing, that tries to call the winner in advance; on the two held-out catalogues it hits Movies but misses Books at the boundary.

2 Related Work

Candidate generation in generative recommendation. Retrieval-augmented generation [9], language-model recommendation [7] and generative retrieval with semantic identifiers [14] all place a nearest-neighbour retrieval step ahead of the generative model, and all optimise the generative stage while treating the item representation as given. Generative retrieval makes the dependence explicit: its identifiers are quantised from a frozen content embedding, so whatever that embedding fails to separate, the codebook cannot recover. Our focus is exactly this taken-for-granted embedding, and

whether a graph-based one is a better choice than the usual text one.

Retrieval for recommendation. Industrial candidate generation is usually a *two-tower* retrieval model trained on behaviour [3, 23] served by approximate nearest-neighbour search [11], or a classic item–item collaborative filter [10]. Our behavioural baseline (Section 6) is exactly that item–item signal; we find it dominates both *content* encoders where interactions are dense, which is why we scope the text-vs-KG comparison to the content, cold-start and long-tail, regime rather than claiming a universal retriever.

Text vs. knowledge-graph item embeddings. The default item embedding is a pretrained sentence encoder [15], or, increasingly, an LLM-based text embedding [18], run over item text. Knowledge-graph embeddings are the content-free alternative: translational and rotational models [1, 17] and graph auto-encoders [8] embed each item from its position in an attribute graph, and knowledge-graph recommenders such as KGAT [19] and KGIN [20] propagate over that graph. These two representations are almost always compared *inside* a trained recommender, where their effect is entangled with the scoring model, the very confound a reproducibility literature warns about [5]. Closest to our question, KG4RecEval [24] asks whether the KG matters at all, but through that same in-recommender route, downgrading or removing the KG inside a trained model. We instead compare text and KG directly as candidate-generation indices, with no scoring model in the loop, and ask which retrieves better and when.

Representation geometry and multimodal fusion. Learned embeddings can be anisotropic, occupying a narrow cone [4, 6]; the effective rank [16] and post-hoc isotropy corrections [12] give tools to quantify and treat this. Multimodal recommenders fuse frozen embeddings over a user–item graph [22, 25, 26], learning weights *after* the embeddings are fixed. We use the geometry tools not to diagnose a broken encoder but to characterise how two healthy encoders differ, and we work one stage earlier than fusion, on the raw retrievability of each representation.

3 Datasets and Representation Geometry

Datasets and encoders. We use six catalogues with disjoint item sets, spanning three themes, four music datasets, plus movies and books from Amazon-KG v2.0 [21] (Amazon interactions linked to a DBpedia item graph), and more than an order of magnitude in knowledge-graph density (Table 1), from KGRec-Music at 44 triples per item down to KKBOX at 2.9. For each item we take two embeddings. The *text* embedding is the output of a pretrained MiniLM sentence encoder [15] (all-MiniLM-L6-v2) over the item’s text description ($d = 384$, unit-normalised). The *KG* embedding is a tuned RotatE [17] knowledge-graph embedding over the item–attribute triples (artist, genre, tag; cast, director, subject; ...), trained per dataset with self-adversarial negative sampling ($d = 256$). A cheap no-training SVD/LSA over the same triples matches or beats it on most datasets (Section 6), and no KG encoder we tried flips a text-vs-KG verdict (Appendix B). The KG embedding never sees item text; the text embedding never sees the graph.

Table 1: Dataset statistics. Six catalogues spanning three themes, music (four), movies and books, the last two from Amazon-KG v2.0 [21] (Amazon interactions linked to a DBpedia item graph). Sparsity is $1 - \text{interactions}/(\text{users} \times \text{items})$; *Rel.* is the number of distinct knowledge-graph relation types and *Tr./item* the mean number of triples per item. Density (*Tr./item*) and relation-type count are independent, MSD-A is dense with only two relation types, Amazon DM sparse with nine, and, as Section 6 shows, neither alone decides the comparison: what matters is whether those relations carry information the item text lacks (KGRec-Music’s social tags do; MSD-A’s artist and genre, already in the item string, do not). The catalogues are ordered by density.

Dataset	Users	Items	Inter.	Sparsity	Triples	Rel.	Tr./item
KGRec-Music	5,199	1,284	130,310	98.05%	56,609	10	44.1
MSD-A	5,970	5,514	33,084	99.90%	165,553	2	30.0
Amazon Movies	5,648	4,779	67,066	99.75%	104,730	20	21.9
Amazon Books	10,485	8,318	115,572	99.87%	66,730	20	8.0
Amazon DM	2,109	1,768	4,873	99.87%	6,619	9	3.7
KKBOX	2,292	2,923	261,238	96.10%	8,341	4	2.9

Are the encoders well-formed? Before comparing what the two encoders retrieve, we check that neither is simply malformed, a collapsed or degenerate embedding would make any retrieval comparison meaningless. We characterise each with four geometry measures, computable in seconds on a catalogue of thousands of items and before any index is built. Let $X \in \mathbb{R}^{N \times d}$ hold the N item embeddings and $\hat{x}_i = x_i / \|x_i\|$.

Four geometry measures. All four are $O(Nd)$ and label-free. *Mean pairwise cosine* $\overline{\cos} = \frac{1}{N(N-1)} (\|\sum_i \hat{x}_i\|^2 - N)$ is near zero for an isotropic cloud, near one for a narrow cone. *Effective rank* $r_{\text{eff}} = \exp(-\sum_k p_k \log p_k)$ over the singular values normalised to sum to one [16], reported against the ambient d , counts the directions actually used. The *centroid-to-residual ratio* $\|\mu\|/\bar{\rho}$ (mean-vector norm over mean residual norm) exceeds one when the shared component outweighs the distinguishing one. *Neighbourhood separation* $\text{sep} = \mathbb{E}_i[s_i^{(10)} - s_i^{(100)}]$, the cosine gap between an item’s tenth and hundredth nearest neighbour, is the operational margin a k -NN index has: a space can pass the first three yet have sep collapse, so we read it as the verdict.

These measures let us establish a baseline fact for the comparison that follows: both the text and the KG encoder are well-conditioned on every dataset (Section 4). Any difference in what they retrieve is therefore a difference in the structure they encode, not an artefact of one being broken.

4 The Two Encoders Are Well-Conditioned

Neither encoder is degenerate. Table 2 reports the geometry of each (both introduced in Section 3). Neither collapses: mean pairwise cosines stay below 0.44, effective ranks use 72–97% of the ambient dimension, and centroid-to-residual ratios are below one. Neighbourhood separation is where they differ, on KKBOX and Amazon DM the KG’s sep (≈ 0.05) is about half the text’s (0.10–0.11), yet the KG still *wins* on KKBOX and loses only on Amazon DM, so a

Table 2: Geometry of the two item encoders across all six catalogues. $\overline{\text{cos}}$ is the mean pairwise cosine, r_{eff}/d the effective rank out of the ambient dimension, $\|\mu\|/\bar{\rho}$ the centroid-to-residual ratio, and sep the neighbourhood separation $\text{sim}@10 - \text{sim}@100$. Both encoders are well-conditioned on every catalogue: neither collapses onto a shared direction, both use most of their dimensions, and both leave a usable neighbourhood margin.

Dataset	Enc.	$\overline{\text{cos}}$	r_{eff}/d	$\ \mu\ /\bar{\rho}$	sep
KGRec-Music	Text	0.283	278/384	0.63	0.073
	KG	0.020	452/512	0.14	0.112
MSD-A	Text	0.441	278/384	0.90	0.055
	KG	0.394	464/512	0.81	0.070
Amazon Movies	Text	0.285	284/384	0.63	0.078
	KG	0.117	459/512	0.36	0.107
Amazon Books	Text	0.252	294/384	0.58	0.085
	KG	0.025	488/512	0.16	0.111
Amazon DM	Text	0.182	287/384	0.47	0.104
	KG	0.002	490/512	0.05	0.047
KKBOX	Text	0.286	290/384	0.64	0.113
	KG	0.002	498/512	0.04	0.051

smaller neighbourhood margin does not by itself decide the contest. We claim only that neither space is *collapsed*, not that geometry is irrelevant.

A second, structural difference (Appendix D, Table 10): text centroids correlate across every pair of catalogues (0.12–0.78), the generic direction one frozen sentence encoder imposes everywhere, while KG centroids are near-orthogonal (−0.09 to 0.07), each fit per corpus, two different kinds of object, a corpus-agnostic map versus a per-corpus fit, not a quality gap. Which helps for retrieval is the next sections’ question.

5 From Geometry to Retrieval

Protocol. Train/test use the per-user hold-out shipped with each dataset (a per-interaction train/test flag), identically for all six catalogues. We build an item–item k -NN index per encoder and, for each user, form a candidate pool as the union of the top- K neighbours of the items in their training history. We then measure

$$\text{POOLRECALL}@K = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathcal{P}_u^K \cap \mathcal{T}_u|}{|\mathcal{T}_u|},$$

the fraction of a user’s held-out items present in the pool. No scoring model is involved, so the measurement isolates candidate generation and attributes any difference to the embedding alone. The pool $\mathcal{P}_u^K$ is the *deduplicated* union of the neighbour lists; we report the random-pool baseline of the same pool size, and use small, operationally relevant budgets $K \in \{5, 10, 20, 50\}$, the range a first stage actually returns before reranking, and one that avoids the pool saturation larger K causes on small catalogues. At matched budget the mean pool size is comparable across encoders on all six catalogues (text vs. KG at $K=10$, within $\sim 10\%$: e.g. 166 vs. 152 on KGRec-Music, 73 vs. 73 on MSD-A), so a win is not a pool-size

artefact, and where the KG wins its pool is if anything the smaller. (Native dimensions differ, text 384, RotatE 256, SVD-KG 512, but all are compared by cosine k -NN at the same budget.) Alongside the two encoders we index their early-fusion concatenation ($\text{text} \oplus \text{KG}$, each part unit-normalised) as a baseline.

Results. Figure 2 shows the encoders per dataset with a tuned RotatE KG and the shipped text, ordered by graph density; the ordering flips with density. On the dense-graph datasets, KGRec-Music (44 triples/item) and MSD-A (30), the KG index leads text at every budget, at $K=10$, POOLRECALL is 0.49 vs 0.30 on KGRec-Music and 0.24 vs 0.05 on MSD-A. On the sparse-graph datasets the order reverses: on Amazon DM (3.7) text reaches 0.20 against KG’s 0.08, and on KKBOX (2.9) the two roughly tie (0.38 vs 0.37). On Amazon Movies and Books, whose text is already the full DBpedia description (not a shipped title), the KG still leads at every budget, 0.20 vs 0.10 and 0.15 vs 0.09 at $K=10$, so their graphs are genuinely complementary, not merely favoured by a thin text. The dense-graph gaps are large and the Amazon DM gap clearly favours text; only KKBOX is a statistical wash (its per-user bootstrap 95% interval on the KG–text difference straddles zero, while the others exclude it). Both encoders stay above the random-pool baseline everywhere, so this is not one representation failing; it is two working representations that suit different catalogues. One caveat carries into the next section: the shipped MSD-A text is a bare, artist-stripped title, so its 0.05 understates what text can do there, Section 6 revisits the comparison with each side’s best-effort representation, and MSD-A is exactly where the picture changes. The **concatenation** baseline is instructive but weak: it tracks whichever single encoder is stronger without *materially* beating it, and its unequal block *dimensions* (text 384, KG 256) implicitly reweight the cosine, sometimes dropping it *below* the winner (KGRec-Music 0.47 vs. the KG’s 0.51). Reciprocal-rank fusion of the two lists appears to do better, but only by enlarging the pool: at a matched candidate budget its advantage vanishes (Section 8), so at equal cost the single-encoder choice stands.

Interpretation. The KG embedding is only as informative as the graph behind it: it helps most when the graph carries signal the text lacks (the Bananarama social tags), and a 3-triple graph is too thin to place an item. Which encoder to index is thus a catalogue property, one Section 6 reads off in advance. Figure 3 makes the mechanism visible on MSD-A, but only partly, the KG’s genre regions are trained-in, so the pool-recall numbers, not the projection, decide.

6 When Do Graphs Win? From Density to Complementarity

Section 5 leaves a practical question: given a new catalogue, which encoder should you index? We first show a tempting one-number answer, graph density, then show it does not survive a fair comparison, and arrive at the predictor that does: whether the graph is *complementary* to the text.

A density rule. Take a strong knowledge-graph embedding, a tuned RotatE [17] over the triples, and each catalogue’s text (the shipped embedding for the four music catalogues, the DBpedia description for Movies and Books). Figure 5 plots the KG-over-text advantage, POOLRECALL@ K of the KG index minus that of

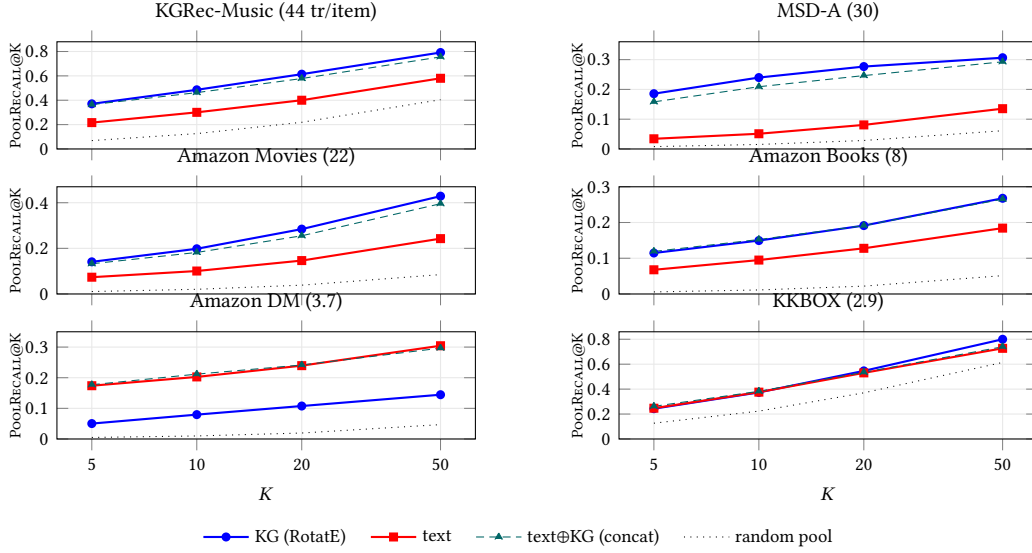


Figure 2: Candidate-pool recall at operational budgets $K \in \{5, 10, 20, 50\}$ for a tuned KG (RotatE), the text encoder, and their early-fusion text⊕KG, across all six catalogues, ordered by knowledge-graph density. For the music catalogues the text is the *shipped* embedding, and the dense→KG ordering it suggests is the tempting rule Section 6 overturns under full-description text (MSD-A then flips to text); for Movies and Books the text is already the full DBpedia description, so their KG lead is genuinely complementary. Concatenation tracks the better single encoder without materially beating it.

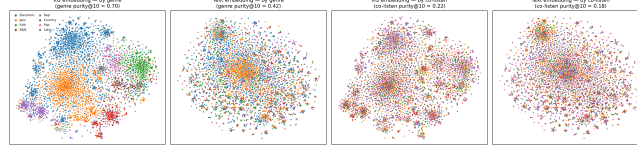


Figure 3: Is the KG neighbourhood really better, or only better on the label it was *trained* on? t-SNE of both encoders on MSD-A. (a, b) coloured by *genre*: the KG forms coherent genre regions, title-only text does not (genre purity@10 0.70 vs. 0.42), but this metric is *circular*, the KG is trained to reconstruct genre. (c, d) the same two maps coloured by a label the KG never saw, *co-listen communities* (KMeans on an SVD of the user-item matrix, purely behavioural): on this non-circular metric the KG's edge nearly vanishes (co-listen purity@10 0.22 vs. 0.18). The genre-coherence that flatters the KG is largely an artifact of the reconstructed label; MSD-A is the redundancy case where the artist-bearing full text wins retrieval (Section 6). The pool-recall numbers, not this projection, decide.

text, against graph density over all six catalogues. The advantage broadly rises with density and changes sign in between, closely enough to tempt a rule: the Pearson correlation between density and advantage is $r = 0.87$ at $K=10$ and $r = 0.86$ at $K=20$. But with $n=6$ the 95% interval on r is roughly $[0.2, 0.99]$; these are descriptive, not inferential, and we lean on them only to state a direction, not a law. On its face this is exactly the actionable rule one wants, count triples per item, index the KG above some threshold, keep text below, and it is the observation that started this study. It is also, we now show, a proxy that masks the real mechanism and can be made to fail, because the text side here is uneven, thin for the music catalogues.

A fair fight breaks the density rule. The crossover above is measured against a *thin* item string, on MSD-A a bare title with the artist stripped out. We give the text side its due: re-encode each

item from its *full* description (artist, title, editorial metadata) with MiniLM and with Qwen3-Embedding (0.6B) [13]. On the KG side we add a trivial baseline, TF-IDF over *all* relation triples followed by truncated SVD, with no training and no content. Table 3 reports recall@10 for every combination over all six catalogues, and two facts stand out. First, this no-training SVD matches or beats the tuned RotatE on three of the six datasets (0.51 vs. 0.49 KGRec-Music, 0.40 vs. 0.37 KKBOX, 0.11 vs. 0.08 Amazon DM; RotatE leads on MSD-A and the two Amazon-KG catalogues): the graph substrate need cost nothing to train, and we return to that in Section 7. Second, and decisively for the density rule, the best-effort ordering is no longer monotone in density. Reading best text against best KG: KGRec-Music (44 triples/item) favours KG by +44%, MSD-A (30) *flips to text* by −6%, Amazon Movies (22) favours KG by +69%, Amazon Books (8) by +48%, Amazon DM (3.7) favours text by −49%,

Table 3: Text vs. KG with best-effort representations on both sides (candidate-pool recall@10), across the six catalogues, ordered by density. Each item is re-encoded from its *full* description with MiniLM and Qwen3-Embedding (0.6B); the KG side gives a tuned RotatE and a cheap no-training SVD/LSA over *all* triples. Δ is best KG – best text with its 95% per-user bootstrap CI (2000 resamples, paired); it *excludes zero on every catalogue*, so each verdict is significant, including the two closest, MSD-A (−0.014, text) and KKBOX (+0.024, KG). Re-selecting the best encoder per side *within* each bootstrap replicate (a winner’s-curse control) leaves the intervals essentially unchanged, e.g. the two tightest stay MSD-A [−.025, −.003] and KKBOX [.017, .031]; on KKBOX the KG win is carried by the SVD encoder, RotatE alone tying the text. Best per row in bold. Density does not order the outcome (Movies, at 22 triples/item, has the largest KG advantage while denser MSD-A favours text); complementarity does. For the two Amazon-KG catalogues the text is the DBpedia title+description.

Dataset	Text (full desc.)		KG		Δ (95% CI)
	MiniLM	Qwen	RotatE	SVD	
KGRec-Music	0.319	0.353	0.486	0.509	+0.156 [.144, .167]
MSD-A	0.254	0.237	0.240	0.209	−0.014 [−.024, −.003]
Amazon Movies	0.101	0.117	0.198	0.174	+0.081 [.070, .092]
Amazon Books	0.095	0.101	0.149	0.105	+0.048 [.042, .055]
Amazon DM	0.203	0.218	0.080	0.111	−0.107 [−.145, −.070]
KKBOX	0.376	0.320	0.374	0.400	+0.024 [.017, .031]

and KKBOX (2.9) by +6%. All six differences exclude zero under a per-user bootstrap, including the two closest, MSD-A and KKBOX (Table 3), so these are real verdicts, not noise. The KG’s biggest win is at 22 triples/item, its clearest loss at 30; the density correlation of Figure 5 leans on a text side that is thin for the music catalogues, and best-effort full text dissolves the ordering. Density alone does not decide the contest.

What decides it: complementarity. The best-effort numbers tell a consistent story once read against *what each graph contains* (Table 1, relation types). KGRec-Music’s graph is built from social tags, mood, style, theme, that no title or editorial blurb states; its content-free SVD embedding beats even the 0.6B Qwen encoder (0.51 vs. 0.35), because it carries information the text simply does not have. KKBOX’s graph, though sparse, adds composer and lyricist, again absent from its “Title – Artist” string, and the KG wins there too. MSD-A is the telling case: its graph is dense (30 triples/item) but its only two relations are artist and genre, *both already present* in the “Artist – Title” text, so the graph is redundant and full text still edges the best KG (0.25 vs. RotatE’s 0.24) despite the density. Movies and Books sharpen the point: their DBpedia graphs carry cast, director, writer and subject that a title-plus-description gloss never lists, so the KG wins on both (0.20 and 0.15 vs. text’s 0.12 and 0.10), and Amazon Movies, at 22 triples per item, is *less* dense than redundant MSD-A at 30 yet the KG wins on Movies and loses on MSD-A, complementarity reversing what density would order. Amazon DM loses for the opposite reason, its graph is simply too

thin (3.7) to place an item, while the text carries a real editorial description. The predictor is not raw density but *complementarity*: the KG wins when its relations encode attributes the text lacks. Density is a necessary ingredient (a graph too thin, like Amazon DM’s, cannot win however novel its relations) but not a sufficient one (a graph dense but redundant, like MSD-A’s, loses), which is exactly why it works as a first screen and fails as a rule.

Density is a real lever, within a fixed relation set (a causal check). That complementarity is the cross-dataset story does not mean density is inert. Holding *everything else* constant isolates its effect. On MSD-A we fix the dataset and the relation types (artist, genre) and vary only the number of triples kept per item, re-embedding each time with the independent SVD encoder and comparing against a fixed text baseline. Figure 4 shows the advantage climbing monotonically from −0.03 at one triple per item, through zero at about two, to +0.14 at the full graph: within a single catalogue, with relation types held constant, more triples do help. Density is a genuine lever on the KG side, it just operates *on top of* the text baseline it is measured against, so whether the lever is enough to overtake text depends on how much the text already knows. (This ablation is measured against the same thin MSD-A text; against the full artist-bearing text the crossover would not be reached, the redundancy point above.)

A second possible confound, interaction sparsity, does not explain the pattern either. On all six points, the KG-over-text advantage tracks KG density ($r = 0.87$) far more than interaction density ($r \approx 0$), again a descriptive contrast, not a test: the KG wins at both high (KGRec-Music, 1.9%) and low (MSD-A, 0.1%) interaction density. What the KG embedding needs is an informative *graph*, not many interactions.

Not a weak text encoder. The complementarity reading is not an escape hatch for an underpowered text encoder. Table 4 re-encodes items with three sentence encoders spanning a $27\times$ range in size, MiniLM (22M), mpnet (110M) and Qwen3- Embedding (0.6B). Their recall stays within ≈ 0.06 on every dataset and no size dominates, the $27\times$ larger Qwen wins on KGRec-Music but loses to tiny MiniLM on KKBOX and MSD-A: scaling the text encoder is not the lever. What moves the text curve is *content*, not parameters, a full description versus a bare title is worth far more than any jump in encoder size, which is why the fair comparison of Table 3 uses each item’s *full* description: on MSD-A that full text (with the artist restored) overtakes the KG, whereas on KGRec-Music no text encoder in the 22M–0.6B range catches the social-tag graph. The deciding factor is the information in the representation, far more than the encoder’s size.

A rule of thumb, offered as a hypothesis. This suggests a two-part heuristic, cheap to try though calibrated on only six points (we test it, not prove it, in Section 8). Count the average triples per item: if it is a handful, the graph is probably too thin, keep text. If it is tens, ask the second question, do those relations carry anything the item text does not already state? Social tags, composer, lyricist, curated attributes: the (cheap SVD) KG embedding is worth trying, it retrieved better and costs less here (Section 7). Artist and genre already in the title: the graph is likely redundant, keep text. Both questions are answered from metadata you already store,

Table 4: Encoder scale is not the lever. Candidate-pool recall@10 under three sentence encoders spanning 27× (MiniLM 22M, mpnet 110M, Qwen 0.6B), each over the item’s full text (text columns match Table 3): the three stay within ≈ 0.06 and no size dominates, Qwen wins on KGRec-Music, tiny MiniLM on KKBOX and MSD-A. Content, not parameters, moves the text curve. Best text encoder per row in bold; the KG column is the tuned RotatE, for reference.

Dataset	Text encoder (full text)			KG
	MiniLM	mpnet	Qwen (0.6B)	
KGRec-Music	0.319	0.312	0.353	0.486
MSD-A	0.254	0.226	0.237	0.240
Amazon Movies	0.101	0.120	0.117	0.198
Amazon Books	0.095	0.080	0.101	0.149
Amazon DM	0.203	0.228	0.218	0.080
KKBOX	0.376	0.353	0.320	0.374

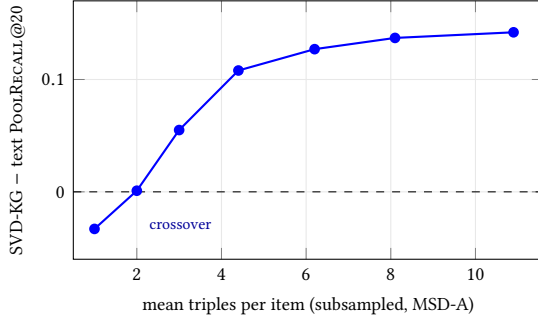


Figure 4: Density is a real lever on the KG side, against a fixed text baseline. On MSD-A we hold the dataset and the relation types (artist, genre) fixed and vary only the number of triples kept per item, re-embedding each time with an independent SVD/LSA knowledge-graph encoder. The advantage over the fixed (title-only) text encoder rises monotonically with triples per item: more triples genuinely help the KG. The zero-crossing at $\sim$ two triples is measured against that thin text baseline, against the full artist-bearing text it would not be reached (Section 6), so this isolates the KG-side mechanism, not the cross-catalogue outcome.

with no trained model and no labelled set. We offer it as a working hypothesis, not a validated rule.

Where content matters: a behavioural baseline. The natural third signal is *behaviour*. An item–item collaborative filter (cosine over the training user–item co-occurrence) dominates *both* content encoders where interactions are plentiful (KKBOX 0.64, KGRec-Music 0.61) but falls below the best content encoder where they are sparse (MSD-A, Amazon DM, both $\approx 0.1\%$ dense; Table 7). This scopes the text-vs-KG question: content is the substrate for the cold-start and long-tail regime, and *there* complementarity decides which content encoder wins. One honest wrinkle: this content-over-behaviour crossover is a *pool-coverage* effect, as a *ranker* the behavioural CF stays marginally ahead even on those sparse catalogues (NDCG@10

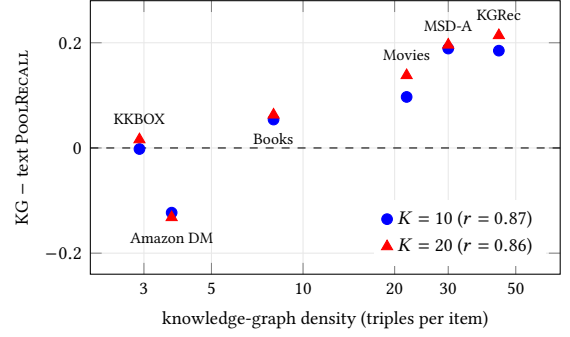


Figure 5: The density rule, with a strong KG. KG (a tuned RotatE) minus text candidate-pool recall against KG density, over all six catalogues; above the dashed line the KG wins. The sign broadly tracks triples per item ($r = 0.87$ at $K=10$, 0.86 at $K=20$, descriptive on six points): dense graphs favour KG, sparse ones text. But the fit is imperfect, Amazon Movies, at 22 triples/item, sits well below denser MSD-A at 30, and it is only the motivating observation, not the law: the text here is the *shipped* embedding for the music catalogues and the DBpedia description for Movies and Books, and Section 6 gives every text side its best-effort full description, whereupon the ordering breaks (MSD-A flips to text), exposing complementarity, not density, as the driver.

0.144 vs. 0.138 on Amazon DM, 0.074 vs. 0.073 on MSD-A; Appendix C), so behaviour orders the reachable set well even where content reaches more of it. The text-vs-KG orderings themselves mostly survive ranking on best-effort representations, though MSD-A’s pool-recall text-lead becomes a statistical tie at NDCG (Appendix C).

7 Why Reach for the Graph

Content-free, and cheap to run. The property we actually demonstrate is that the KG substrate is *content-free*: it reads no item text, sidestepping the coverage and language gaps of text, and needs no large pretrained encoder. The SVD-KG encoder of Section 6 is latent semantic analysis of the item–attribute matrix, a classic content-based factorisation; the novelty is the head-to-head, not the method. It is also cheap to *run*: on MSD-A (5,514 items, one CPU) it encodes at 256 items/s vs. 46 for MiniLM, reading no content. But the symmetric cheap baseline must be stated: a no-training TF-IDF+SVD on the item *text* is just as cheap, and stays within a few points of the neural encoder (Table 9). So the practical contest is cheap-vs-cheap, SVD-text vs. SVD-KG, and the verdict is unchanged: SVD-KG wins where the graph is complementary, SVD-text on Amazon DM (MSD-A a near-tie). We do not over-claim *low cost*: the timing counts encoder inference, not the construction of the attribute graph (on Amazon-KG an external DBpedia linkage), and the 5.5× narrows on GPU. *Content-free* holds unconditionally; the cost saving is real but only for run-time inference.

A reranker inherits the pool’s ceiling. A tempting objection is that a strong reranker downstream will wash out the retrieval difference. It cannot: a reranker only reorders the pool it is given, so its recall is

Table 5: Which retrieved context helps? recall@10 over all 1,756 recoverable KGRec-Music users, each of four contexts scored by a deterministic, reproducible similarity proxy (rank the pool by the context embedding’s similarity to the user’s history), with 95% bootstrap CIs. Fusing the text description with the KG triples wins; text and KG alone overlap. Text and graph are *complementary*, each carries signal the other lacks. Pool ceiling 0.371.

Context shown per candidate	recall@10	95% CI
none (bare id; random 10-of-40)	0.093	,
text description	0.189	[0.180, 0.198]
KG triples (social tags)	0.194	[0.184, 0.203]
text + triples	0.228	[0.219, 0.237]

capped by the pool’s *ceiling*, the fraction of held-out items in the top- M stage-1 list. No reordering recovers what stage-1 never returned, so the choice of retriever is a first-order decision the generator inherits. This bites hardest for agentic recommenders: from a pool that never contained the answer an LLM still returns ten fluent, plausible items, and no metric on that output reveals a different pool was reachable. Which context to feed the generator, once the pool is fixed, is the separate, and more measurable, question we take up next.

What to put in the generator’s context: a RAG view. Once the pool is fixed, a retrieval-augmented recommender still chooses *what to say about each candidate*, the prompt’s *context* block. Holding the KGRec-Music KG pool fixed, we vary that block over four contexts about each candidate:

```
Candidate {id}: {CONTEXT}
CONTEXT ∈ { nothing (bare id); the item’s text description; its raw KG triples
“genre: . . . ; tag: . . . ”; both }.
```

The candidate set is identical for every context, the KG stage-1 pool, and the 1,756 “recoverable” users are those whose held-out item lands in that pool, so the selection is the same across contexts and only conditions the absolute numbers on KG-reachable items. We score each context by a *deterministic* similarity proxy, ranking the pool by the context embedding’s cosine to the user’s history, no language model in the loop, over all 1,756 users, with 95% bootstrap CIs (Table 5). Any informative context beats none (0.09, random), and the ordering is decisive and significant: fusing the text description with the KG triples wins (0.228, CI [0.219, 0.237]), clearly above either alone (KG 0.194, text 0.189, overlapping intervals). Text and graph are *complementary*, each carries signal the other lacks, the retrieval-side story of Section 6 reappearing at the context step. Where the graph merely restates the text (MSD-A), the contexts coincide. The measure is a similarity proxy, not a live generator, so we claim only this: the context that best distinguishes the right candidate is the one the text lacks, on a complementary catalogue, both.

8 Discussion

A cheap a-priori heuristic (not a validated predictor). The story is only useful if “does the graph add what the text lacks?” can be estimated *before* indexing, without retrieval or labels. We define a

redundancy score that tries. Its lexical ingredient, the mean fraction of a triple’s surface tokens already in the item’s text (0.11 KGRec-Music, 0.20 KKBOX, 0.19 MSD-A, 0.80 Amazon DM), tracks the advantage ($r = 0.75$) but mis-ranks MSD-A, whose genre is implied by the artist though not literally present. A learned ingredient catches that: a 5-fold cross-validated linear probe predicting the (SVD) triple structure from the frozen text embedding, whose R^2 (a modest 0.21 at its highest) scores MSD-A the most redundant of the four, correctly. Taking the stronger of the two, the KG-winners (0.11, 0.20) and text-winners (0.21, 0.80) fall either side of a threshold near 0.2. On two held-out catalogues the score is 0.14 (Movies) and 0.21 (Books): Movies, well below the threshold, is called correctly (the KG wins), but Books, scoring exactly MSD-A’s 0.21 on the text-winning side, favours the KG anyway ($\Delta = +0.048$, Table 3). Out of sample, then, one clean hit and one boundary *miss*, not a clean 2/2: a threshold read off four points makes this an *encouraging heuristic to try*, not a validated predictor, but one with a mechanism (redundancy) a bare density number lacks. Movies is the sharp case, at 22 triples/item it is *less* dense than redundant MSD-A (30) yet its cast-and-subject graph is complementary, so the KG wins where density would have predicted text.

Content, not format. By *content-free* we mean the KG reads no item text, not that it carries no information: its triples are the attributes the text omits. Every KG win here coincides with facts absent from the text, and the one same-information case (MSD-A) has text winning, so we measure information *complementarity*, not graph *structure* per se; isolating format (paraphrasing triples to prose and re-embedding) is future work.

Fusion is not a free lunch at equal budget. The two encoders retrieve partly complementary items, so fusing them is tempting. Early-fusion concatenation barely helps and sometimes hurts (KGRec-Music 0.47 vs. the KG’s 0.51). Reciprocal-rank fusion [2] of the two k -NN lists looks much better, +0.06 to +0.13 pool recall@10, but the gain is an artefact of *budget*: fusing two top- K lists nearly doubles the pool (283 vs. 167 items on KGRec-Music), and at a matched pool size it collapses below the best single encoder, so at equal cost the single-encoder choice stands. (Context-level fusion is different: conditioning a generator on both text and triples does help, Table 5, because there the budget is one candidate, not a pool.)

Toward generative retrieval. The same logic reaches generative retrieval, whose semantic-ID identifiers [14] quantise a frozen content embedding: the codebook can only group items by structure that embedding already separates. Where the graph is complementary, quantising the KG rather than the text is a natural next step, untested here.

9 Conclusion

Comparing text and knowledge-graph embeddings for candidate generation across six catalogues, with no scoring model in the loop, the answer is not *graphs beat text*: under best-effort representations the two are roughly even, and *complementarity*, not density, decides, the KG winning only when its relations encode what the text lacks, and a no-training SVD over the triples is as cheap as its text counterpart, so the contest is cheap-vs-cheap. We are deliberate about scope: the six points are a direction, not a law; the redundancy score a working hypothesis, not a validated predictor; cold-start is proxied by interaction density, not a cold-item split; and we evaluate retrieval, not a scorer. Within those bounds it maps *when*, and with which encoder, content helps candidate generation.

References

- [1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*, Vol. 26. 2787–2795.
- [2] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *Proceedings of the 32nd International ACM SIGIR Conference*. 758–759. doi:10.1145/1571941.1572114
- [3] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *ACM Conference on Recommender Systems (RecSys)*.
- [4] Kavin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 55–65.
- [5] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys)*. 101–109. doi:10.1145/3298689.3347058
- [6] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation Degeneration Problem in Training Natural Language Generation Models. In *International Conference on Learning Representations*.
- [7] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*. 299–315. doi:10.1145/3523227.3546767
- [8] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. In *NeurIPS Workshop on Bayesian Deep Learning*.
- [9] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, Vol. 33. 9459–9474.
- [10] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing* 7, 1 (2003), 76–80.
- [11] Yu A. Malkov and Dmitry A. Yashunin. 2020. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 4 (2020), 824–836. doi:10.1109/TPAMI.2018.2889473
- [12] Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-Top: Simple and Effective Postprocessing for Word Representations. In *International Conference on Learning Representations*.
- [13] Qwen Team. 2025. Qwen3 Embedding. <https://huggingface.co/Qwen/Qwen3-Embedding-0.6B>.
- [14] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. Recommender Systems with Generative Retrieval. In *Advances in Neural Information Processing Systems*, Vol. 36.
- [15] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 3982–3992.
- [16] Olivier Roy and Martin Vetterli. 2007. The Effective Rank: A Measure of Effective Dimensionality. In *15th European Signal Processing Conference (EUSIPCO)*. 606–610.
- [17] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations (ICLR)*.
- [18] Liang Wang, Nan Yang, Xiaolong Huang, et al. 2024. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *arXiv preprint arXiv:2212.03533* (2024).
- [19] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 950–958. doi:10.1145/3292500.3330989
- [20] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhengguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning Intents behind Interactions with Knowledge Graph for Recommendation. In *Proceedings of the Web Conference 2021 (WWW)*. 878–887. doi:10.1145/3442381.3450133
- [21] Yuhang Wang, Qing Xie, Mengzi Tang, Lin Li, Jingling Yuan, and Yongjian Liu. 2024. Amazon-KG: A Knowledge Graph Enhanced Cross-Domain Recommendation Dataset. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. doi:10.1145/3626772.3657880
- [22] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal Graph Convolution Network for Personalized Recommendation of Micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1437–1445. doi:10.1145/3343031.3351034
- [23] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, et al. 2019. Sampling-Bias-Corrected Neural Modeling for Large Corpus Item Recommendations. In *ACM Conference on Recommender Systems (RecSys)*.
- [24] Haonan Zhang, Dongxia Wang, Zhu Sun, Yanhui Li, Youcheng Sun, Huizhi Liang, and Wenhai Wang. 2025. KG4RecEval: Does Knowledge Graph Really Matter for Recommender Systems? *ACM Transactions on Information Systems* (2025). doi:10.1145/3713071
- [25] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. 2021. Mining Latent Structures for Multimedia Recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3872–3880. doi:10.1145/3474085.3475259
- [26] Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. 2023. Bootstrap Latent Representations for Multimodal Recommendation. In *Proceedings of the ACM Web Conference 2023*. 845–854.

A Generative AI Usage Disclosure

In accordance with the call for papers, we disclose the use of generative AI. All quantitative results were produced by the released scripts run on the authors’ data, not generated or estimated by a language model, and every number in the tables, figures and text was read back from those script outputs and checked by the authors. No language model sits in any measurement loop: in particular, the four-way context ablation of Table 5 uses a *deterministic* similarity proxy over all 1,756 users. The text and knowledge-graph embeddings, the graph triples, the candidate pools, and the datasets are the authors’ own. A language model was used only for general spelling and grammar corrections. The authors take full responsibility for the correctness of the claims and for the final content.

Code and data availability. All experiment scripts, geometry audit, PoolRecall, the SVD/RotatE knowledge-graph encoders, the cost and ranking evaluations, the 1,756-user context ablation, and the movies-and-books Amazon-KG v2.0 catalogues, will be released in a public repository on acceptance. The music datasets are public benchmarks; the movies and books data is Amazon-KG v2.0 [21] with item text fetched from the public DBpedia SPARQL endpoint, so the full pipeline is reproducible from released code without private assets.

B Robustness to the Knowledge-Graph Encoder

A natural objection is that the text-vs-KG verdict might hinge on our particular KG encoder (a tuned RotatE). It does not. Table 6 evaluates four KG encoders on the four music catalogues under the identical recall@10 protocol of Section 5: TF-IDF and BM25 truncated-SVD over the raw triples, and RotatE and ComplEx trained with self-adversarial negative sampling ($d=256$, 100 epochs, 16 negatives). (On the two Amazon-KG catalogues we report SVD and RotatE, Table 3, which already agree on the verdict.)

Two conclusions hold. First, the *training-free* SVD is the best KG encoder on three of the four music catalogues; the trained models do not dominate the cheap linear factorisation, RotatE edges it only on MSD-A, whose dense two-relation graph suits a relational model, ComplEx trails everywhere, and a TransE we also tried collapses at ≤ 0.04 (its translational geometry does not yield a cosine-kNN-friendly item space). Second, reading each side’s *best* encoder, the text-vs-KG verdict is stable: KGRec-Music favours the KG under all four encoders, MSD-A and Amazon DM favour text under all four (the best KG on MSD-A, RotatE 0.240, stays under text’s 0.254). KKBOX is the one encoder-dependent case: the SVD encoders win for the KG (0.400/0.390) while RotatE (0.374) ties the text (0.376)

and ComplEx (0.308) trails it, so its KG verdict rests on the cheap SVD, not the trained models. Which representation wins is set by *complementarity* (Section 6) far more than by the choice of KG encoder.

Table 6: Robustness to the knowledge-graph encoder (recall@10, identical item–item kNN protocol). Four KG encoders: TF-IDF and BM25 truncated-SVD over the triples, and RotatE / ComplEx trained with self-adversarial negative sampling ($d=256$, 100 epochs). Best KG per row in bold. The text-vs-KG verdict (last column) is invariant to the encoder; KGRec-Music and KKBOX favour the KG under every encoder; MSD-A and Amazon DM favour text under every encoder, even the best KG on MSD-A (RotatE, 0.240) stays under text’s 0.254. The training-free SVD is the best KG encoder on three of the four datasets.

Dataset	text (full)	knowledge-graph encoder				verdict
		SVD	BM25	RotatE	ComplEx	
KGRec-Music	0.319	0.509	0.465	0.486	0.325	KG>txt
KKBOX	0.376	0.400	0.390	0.374	0.308	KG>txt
MSD-A	0.254	0.209	0.233	0.240	0.194	txt>KG
Amazon DM	0.203	0.111	0.104	0.080	0.074	txt>KG

C Ranking Metrics and the Behavioural Baseline

Table 7 gives the behavioural baseline referenced in Section 6: an item–item collaborative-filtering retriever versus the best-effort content encoders, in candidate-pool recall, ordered by interaction density. Behaviour dominates where interactions are dense and falls below content where they are sparse. Interaction density is not the whole story, though: Amazon DM and Books share the same 0.13% density yet fall on opposite sides (content beats CF on Amazon DM, 0.218 vs. 0.199; CF beats content on Books, 0.214 vs. 0.149), because content wins only when the text itself is rich, Amazon DM ships a full editorial description while Books carries a short DBpedia gloss. What content competes against behaviour is the richness of the text, gated by low interaction density, not density alone.

Two caveats. Our catalogues hold 1.7k–8.3k items, so the pool a user’s history induces covers a sizeable fraction of the catalogue; whether these orderings persist at industrial scale (millions of items) is an extrapolation we do not test. And the cold-start regime is operationalized by low dataset-level *interaction* density (Amazon DM, MSD-A), not by a held-out cold-item split, the direct test we leave to future work.

Our headline metric is candidate-pool recall, which measures whether a held-out item *reaches* the pool; a reviewer rightly asks whether that translates into ranking quality. Table 8 re-runs each signal as a *ranker* on the *same best-effort representations* as Table 3 (full-text MiniLM, not the shipped title), reporting NDCG@10. Two things hold. First, the best-effort text-vs-KG verdict mostly survives the move to ranking, text leads on Amazon DM, the KG on the complementary catalogues. MSD-A is the sharpest read of *redundancy*: the two *tie* at NDCG (0.073 vs. 0.072; paired-bootstrap CI on the difference $[-.008, +.010]$ straddles zero), which is exactly

what a redundant graph predicts, at ranking the KG neither improves nor degrades the text, it is inert (its pool-recall text-lead is the finer effect). The shipped title-only text collapsed to 0.004 here, understating text far more; the fair comparison matters as much for ranking as for coverage. Second, the behavioural CF is the strongest ranker on five of six catalogues, *including* the sparse Amazon DM and MSD-A where content leads in *pool* recall (there CF stays marginally ahead at NDCG, 0.144 vs. 0.138 and 0.074 vs. 0.073). The content-vs-behaviour crossover is thus sharper in pool coverage than in ranking: behaviour orders the reachable set well even where content reaches more of it, which tempers, but does not remove, the content advantage in the cold-start regime.

(Movies sharpens the pool-vs-ranking gap the other way: the CF has the higher pool recall, 0.235 vs. the KG’s 0.198, yet the KG ranks the reachable items better, NDCG 0.047 vs. 0.040.)

Table 7: A behavioural baseline. Candidate-pool recall@10 of an item–item collaborative-filtering retriever (cosine over the train user–item co-occurrence) against the best-effort *content* encoders (best text, best KG from Table 3), over all six catalogues, ordered by *interaction* density. Behaviour dominates on the higher-interaction catalogues; content overtakes only in the sparsest, cold-start regime (Amazon DM, MSD-A). Best per row in bold.

Dataset	inter. dens.	CF	best text	best KG
KKBOX	3.9%	0.641	0.376	0.400
KGRec-Music	1.9%	0.612	0.353	0.509
Amazon Movies	0.25%	0.235	0.117	0.198
Amazon Books	0.13%	0.214	0.101	0.149
Amazon DM	0.13%	0.199	0.218	0.111
MSD-A	0.10%	0.194	0.254	0.240

Table 8: Ranking metrics on the *same best-effort representations* as Table 3 (full-text MiniLM, tuned RotatE, no-training SVD, item–item CF): each signal used as a *ranker* (score every item by max similarity to the user’s history, take the global top-10), NDCG@10. The best-effort text-vs-KG verdict mostly survives ranking: text leads the KG on Amazon DM (significant), the KG on the complementary catalogues; on MSD-A the two *tie* at NDCG (0.073 vs. 0.072; paired per-user bootstrap CI on the text–KG difference $[-.008, +.010]$ straddles zero), exactly what a *redundant* graph predicts: the KG is inert at ranking, it neither helps nor hurts. The shipped title-only text collapsed to 0.004 here, understating text far more. Behavioural CF is the strongest ranker on five of six catalogues (it edges text on Amazon DM, 0.144 vs. 0.138). Best per row in bold.

Dataset (NDCG@10)	CF	text	RotatE	SVD-KG
KGRec-Music	0.226	0.054	0.109	0.118
KKBOX	0.128	0.012	0.016	0.020
Amazon Movies	0.040	0.030	0.047	0.038
Amazon Books	0.045	0.020	0.038	0.015
Amazon DM	0.144	0.138	0.036	0.063
MSD-A	0.074	0.073	0.072	0.057

A symmetric no-training baseline. Our cheap KG encoder is a no-training TF-IDF+SVD over the triples (Section 7). Its mirror on the text side, the same factorisation over the item *text*, lets us strip the neural encoder from *both* sides and test complementarity in the cleanest form. Table 9 reports it: each SVD roughly matches its neural counterpart, and the no-training head-to-head keeps the verdict, SVD-KG where the graph is complementary, SVD-text on Amazon DM, MSD-A a near-tie, so the outcome is set by complementarity, not by encoder quality on either side.

Table 9: Cheap vs. cheap. Candidate-pool recall@10 of the two no-training SVD encoders, TF-IDF+truncated SVD over the item *text*, and over the KG *triples*, each with its neural counterpart (MiniLM, RotatE) for reference. Each SVD is within a few points of its neural counterpart, most often slightly below (SVD-text trails MiniLM on four of six catalogues, SVD-KG trails RotatE on three), so the head-to-head runs with no training on either side. The point is that the text-vs-KG verdict is unchanged: stripped to cheap-vs-cheap the SVD-KG still wins where the graph is complementary, SVD-text on Amazon DM, with MSD-A now a near-tie. Complementarity, not encoder quality, decides. Best no-training encoder per row in bold.

Dataset	Text		KG	
	SVD	MiniLM	SVD	RotatE
KGRec-Music	0.337	0.319	0.509	0.486
MSD-A	0.201	0.254	0.209	0.240
Amazon Movies	0.073	0.101	0.174	0.198
Amazon Books	0.070	0.095	0.105	0.149
Amazon DM	0.214	0.203	0.111	0.080
KKBOX	0.330	0.376	0.400	0.374

D Cross-Corpus Encoder Geometry

Table 10 gives the cross-corpus centroid cosines summarised in Section 4 (text shares a generic direction across all six disjoint catalogues; the KG centroids are near-orthogonal, each fit per corpus).

Table 10: Cross-corpus cosine between item-encoder centroids, over the six catalogues (disjoint item sets). Upper triangle: Text; lower triangle: KG (RotatE). Text centroids share a generic direction on every pair (0.12–0.78), the signature of one frozen sentence encoder applied everywhere, while KG centroids are essentially orthogonal (−0.09 to 0.07), each embedding fit separately on its corpus. The contrast marks two different kinds of representation, not a quality gap.

	KGRec	MSD-A	Movies	Books	Amz DM	KKBOX
KGRec-Music	1.00	0.52	0.31	0.31	0.68	0.76
MSD-A	−0.05	1.00	0.12	0.20	0.64	0.52
Amazon Movies	−0.06	−0.01	1.00	0.57	0.45	0.49
Amazon Books	0.07	0.01	−0.03	1.00	0.51	0.44
Amazon DM	−0.03	0.07	0.01	0.03	1.00	0.78
KKBOX	−0.02	−0.09	0.03	0.01	0.04	1.00