Argumentative Graph-RAG for Participatory Democracy: When Graph Structure Helps Retrieval, and Why

Michele Pagano Mariano[1,2], Rafael Angarita[2,3] and Nicolas Maudet[2]

[1]Politecnico di Torino, Turin, Italy
[2]LIP6, Sorbonne University, Paris, France
[3]Paris Nanterre University, Nanterre, France

Abstract

Participatory democracy platforms produce large volumes of citizen proposals, making it hard to identify redundant or conflicting contributions. Prior work can recover the argumentative structure of such debates—support, contradiction, equivalence—and organize it into an argument graph; whether that structure then improves retrieval over embedding similarity alone is less settled. We evaluate a Graph-RAG approach that generates its candidate pool by traversing such a graph, against a dense baseline and a diversification ablation, on conflict-map retrieval and debate summarization over 30 policies. On a native retrieval pool we find no advantage for graph structure: it is easy enough that even random selection is competitive. On an adversarially hardened pool seeded with sibling-policy distractors the ranking reverses, but not for the usual reason. The vulnerability lies in the diversification step, which is drawn toward the injected distractors; removing it, with no graph signal at all, resists them just as well. What the graph contributes is a cleaner candidate pool upstream, letting the system stay distractor-resistant while still returning a varied, non-redundant set rather than a narrow, repetitive one. This protection is contingent on how distractors are wired: when they connect among themselves, traversal is drawn into them and the advantage largely disappears. We suggest treating pool difficulty and distractor wiring as explicit variables in Graph-RAG evaluation.

Keywords

Graph-RAG, Retrieval-Augmented Generation (RAG), Argument Mining, Knowledge Graphs, Participatory Democracy

CEUR-WS.org/Vol-4312/paper11.pdf

1. Introduction

Participatory democracy platforms such as Decidim [1], Cap Collectif [2], and Consul [3] enable thousands of citizens to propose and debate public policy. Their adoption has been substantial: Decide Madrid, built on Consul, saw over 430,000 users submit more than 26,000 proposals and 125,000 comments in its citizen-proposals process alone [3]. Similar platforms are now deployed by city and national governments across Europe and Latin America for participatory budgeting, collaborative law-making, and large-scale public consultations such as France's Grand Débat National.

Their success creates a problem of scale. As the number of contributions grows, no citizen can read the full debate, so they cannot tell which existing proposals resemble or oppose their own, and no organizer can manually review thousands of comments to determine where consensus or disagreement lies. This is precisely the difficulty documented on Consul-based platforms, where both citizens and administrators reported being unable to obtain an overview of the volume of proposals and comments generated [3]. The result is a form of information overload that risks undermining the very inclusiveness these platforms are meant to provide, since a citizen's contribution is only meaningfully heard if it can be situated within the broader debate.

Prior work has shown that some of this structure can be recovered automatically. Natural language processing models can detect argumentative relations between citizen proposals, determining whether two proposals make the same point or whether they are in direct conflict. These relations can be

RAGE-KG 2026: The Third International Workshop on Retrieval-Augmented Generation Enabled by Knowledge Graphs, co-located with ISWC 2026, October 25–26, 2026, Bari, Italy
*Corresponding author.
michele.paganomar@gmail.com (M. P. Mariano); rafael.angarita@lip6.fr (R. Angarita); nicolas.maudet@lip6.fr (N. Maudet)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-04

organized into an argumentative graph that makes the shape of the debate explicit rather than leaving it buried in unstructured text [4]. Helping a user see this shape—the relevant arguments on both sides of an issue, without redundancy—is fundamentally a retrieval problem: given a policy under debate, which arguments should be surfaced, and how should they be organized so that the retrieval itself does not reintroduce the very redundancy and imbalance it is meant to resolve?

This paper investigates whether an argument graph, once recovered, is useful at retrieval time. We study this primarily through conflict-map retrieval—surfacing a short, stance-balanced set of arguments a moderator can read directly—and additionally test whether any gains propagate to a downstream debate-summarization task. We compare a graph-structure-aware retriever against a dense baseline and a diversification ablation, holding corpus, embedding model, and diversification mechanism fixed so that a difference between two systems isolates the component that separates them.

Our findings are threefold. First, on the native pool—each policy's own curated arguments, which are relevant almost by construction—no method separates from another, the graph included; even random selection is competitive. This is less a result about retrieval than about the pool: when nearly everything is relevant, there is no headroom for any signal to show through. Second, on an adversarially hardened pool seeded with topically adjacent distractors, the systems finally separate, and the graph retriever overtakes the dense baseline it had trailed. But our ablation locates the cause in the diversification step rather than in the graph: removing diversification entirely, with no graph signal at all, resists the distractors just as well. Third, the graph's resistance to distractors depends on how they sit in the graph. When they attach only to the genuine arguments, graph traversal never reaches them and the resistance holds; but when they also connect to one another, they form their own region that the traversal is drawn into, and it disappears. The graph resists distractors, in other words, only when the distractors are structurally isolated—so pool difficulty and distractor wiring are worth treating as explicit variables in Graph-RAG evaluation, not fixed at a convenient default.

2. Related Work

Graph-RAG. Combining knowledge graphs with retrieval-augmented generation has become a common strategy for grounding LLM outputs in structured domain knowledge. Edge et al. [5] build community summaries over an extracted knowledge graph to support query-focused summarization at corpus scale; subsequent systems such as LightRAG [6] and HippoRAG [7] pursue similar gains through lighter-weight indexing and memory-inspired retrieval respectively. These systems typically report end-to-end gains, but the graph-augmented pipeline usually differs from its baseline in more than the graph signal alone, so it is not always clear how much of the gain the graph itself explains. This motivates the single-variable design of the present paper, which holds every component fixed except the retrieval signal; as our results show, isolating the graph in this way can leave its apparent benefit looking rather different from what an end-to-end comparison would suggest.

Argument mining and retrieval. Argument retrieval has an established evaluation tradition, notably the Touch'e [8] and PerspectiveArg [9] shared tasks, both scoring systems on graded relevance to a controversial topic. The IBM Project Debater line of work, including the argument-quality corpus of Gretz et al. [10] we build on, provides large-scale stance-labeled arguments. Argumentative relations between arguments have been identified both by supervised learning over sentence pairs [11] and by grounding them in textual entailment [12, 13]. We adopt the latter framing: our CONTRADICTS and EQUIVALENT edges are recovered by a supervised NLI cross-encoder rather than by embedding similarity (Section 4.1), since the two capture different notions of relatedness.

Diversification and adversarial pool construction. Maximal Marginal Relevance [14] remains the standard mechanism for balancing relevance against redundancy in a retrieved set, with intra-list distance [15] and α-nDCG [16] the standard ways to measure the result; recent work continues to treat diversity as a first-class objective in RAG pipelines [17]. Closest to our design, Cuconasu et al. [18]

show that native, largely-relevant pools understate retrieval difficulty and recommend constructing hard-negative pools instead. We adopt this and extend it: for a graph retriever, the *wiring* of injected distractors—not merely their presence—determines whether they are resisted (Section 6.4).

3. Retrieval over an Argument Graph

All systems we study in this paper are stance-aware: for a given policy, each returns a top-k list of pro arguments (supporting the policy) and a top-k list of con arguments (opposing the policy). They share a common retrieval framework (Section 3.1) and differ in one component at a time — the candidate pool, the relevance signal, or the diversification step — so that a difference between two systems can be read as the effect of the component that separates them.

3.1. Shared Retrieval Framework

Every system selects its top-k list from a candidate pool R using Maximal Marginal Relevance (MMR) [14], which builds the list greedily, at each step trading a candidate's relevance against its similarity to what is already selected. Letting $S \subseteq R$ be the set already selected, the next item added is

$$\arg\max_{d_i \in R \setminus S} \Big[\lambda \cdot \mathrm{rel}(d_i) - (1-\lambda) \max_{d_j \in S} \mathrm{sim}(d_i, d_j) \Big], \tag{1}$$

where $\mathrm{rel}(d_i)$ is the relevance score (defined per system below) and the second term penalizes resemblance to the closest item already chosen, spreading the set out rather than filling it with near-duplicates. We fix the weight λ=0.5 for every system, so any diversity difference in Section 5.1 follows from the candidate pool R and the relevance signal $\mathrm{rel}(\cdot)$, not from re-tuning MMR. These two — how R is generated and what $\mathrm{rel}(\cdot)$ ranks it — are what distinguish the systems compared here.

3.2. PR-GraphRAG: Retrieval by Graph Traversal

Dense retrieval builds its candidate pool from embedding proximity, which cannot separate an argument that answers *this* policy's question from one that lives in the same topic and shares its vocabulary while answering a different question merely shares its topic's vocabulary. On a corpus where policies have topically adjacent siblings—two competing zoning proposals for the same neighborhood, say—an argument for one sibling sits as close to the target policy as a genuinely relevant argument does, and nothing in the embedding geometry tells them apart. The argument graph, however, already records the distinction the embedding space does not: an argument attaches to a policy through an explicit ATTACKS edge and to opposing arguments through CONTRADICTS edges. Generating candidates by following those edges constrains the pool by argumentative provenance rather than lexical proximity.

PR-GraphRAG (PageRank-seeded Graph-RAG) does this in two stages:

- *Candidate generation*: the pool is seeded by the arguments holding a direct attacks edge to the target policy, then expanded one hop along contradicts edges — a con argument that contradicts a seeded pro is added to the con pool, and symmetrically for pro. The expansion admits arguments that genuinely rebut the debate, whether or not they sit close to the policy centroid in embedding space.
- *Relevance signal*: candidates are ranked by their PageRank [19] in the policy graph, min–max normalized per pool, so $\mathrm{rel}(d_i) = \widehat{\mathrm{PR}}(d_i)$. PageRank is computed once, over the full graph, so every system sees the same centrality scores.

The first item selected is therefore the most central argument in the policy's rebuttal structure, rather than the one nearest the pool centroid. One property of this design matters more than we anticipated: because the pool comes from graph traversal, it is filtered *before* MMR runs, and an argument with no edge into the target policy's neighborhood is never a candidate at all. Sections 5.1 and 6 trace the behavior we observe under distractor pressure back to this fact.

3.3. Baselines and Ablations

We compare PR-GraphRAG against three further systems, holding corpus, embedding model, and k fixed across all four

- **StandardRAG** (embedding baseline): the candidate pool is the policy's own native argument set, therefore every argument already carrying a supports or attacks edge to the policy capped at 200 per stance. Relevance in Eq. (1) is each candidate's cosine similarity to the *centroid* of that pool ($\text{rel}(d_i) = \text{sim}(d_i, \bar{d})$, with $\bar{d}$ the pool mean). Ranking by proximity to the centroid, rather than to the raw query, favors arguments central to the retrieved set once an initial relevant pool has been established. No graph signal is used at any stage. This is the direct comparison for **PR-GraphRAG**: it differs in candidate generation and relevance signal, with corpus, embedding model, MMR mechanism, and k all held fixed.
- **RAG-no-MMR** (diversification ablation): the identical candidate pool and relevance signal as **StandardRAG**, but taking the top-k directly instead of re-ranking with MMR. Holding pool and signal fixed, this isolates the effect of the MMR step itself — the one system that varies the diversification component rather than the pool or the signal.
- **Random**: uniform random selection from the candidate pool — a floor for what any relevance signal should beat, and, as Section 6 notes, a diagnostic for whether the pool is discriminative at all.

4. Methodology

4.1. Corpus and Argument Graph

We use the IBM-ArgQ-Rank-30k corpus [10]: approximately 30,500 arguments across 71 policies, each labeled by stance (pro/con) relative to its policy. From it we build a policy–argument graph with three edge types. ATTACKS records an argument's stance relation to its policy, taken directly from the corpus labels; CONTRADICTS records pairwise stance opposition between arguments; and EQUIVALENT records near-paraphrase relations between arguments making the same point. The latter two are recovered by an NLI cross-encoder[1] over within-policy argument pairs rather than by embedding similarity, since embedding distance conflates topical proximity with argumentative opposition. CONTRADICTS edges drive graph-aware retrieval's multi-hop expansion; EQUIVALENT edges support the community detection used for subtopic-level analysis.

Raw NLI scores over-connect the graph—on a bipartite pro/con structure almost every pro trivially "contradicts" almost every con—so we keep only high-confidence, bidirectionally-agreed CONTRADICTS edges. This cuts the edge count from roughly 1.06M (33% of possible pairs) to 344K (9%), a level at which PageRank centrality and multi-hop traversal reflect genuine argumentative structure rather than near-universal connectivity.

4.2. Use Cases

We evaluate retrieval through two downstream use cases, chosen to reflect the two ways a moderator or citizen actually consumes retrieved arguments: scanning a short, high-precision list, and reading a longer synthesized account.

UC1: Conflict-Map Retrieval. Given a policy, return a short, stance-balanced set of pro and con arguments that a moderator can scan directly to see where a debate's genuine points of disagreement lie. This is precision-first: the set is read in full, so every irrelevant or redundant item has a direct cost, and diversity matters as much as relevance. We evaluate UC1 at candidate-pool size 200 and report

[1] `cross-encoder/nli-deberta-v3-base`; labels read from `config.id2label` by name, not index, to avoid a silent permutation error.

primarily at k=5, a deliberately small cutoff reflecting that a conflict map is meant to be read in full rather than skimmed (Section 5.1).

UC2: Debate Summarization. Given a policy, generate a short natural-language summary of the debate, covering the main pro and con positions in balance, grounded in the retrieved arguments. This is recall-oriented and generation-downstream: retrieval quality matters here not because a person reads the retrieved list directly, but because it bounds what the summarizer can faithfully say. We evaluate UC2 at k=15, a wider budget appropriate for feeding a summarization prompt (Section 5.2).

Both use cases share the same retrieval systems, corpus, and candidate-pool construction; they differ only in k, in whether the retrieved set is read or summarized, and in their metrics (Section 4.6).

4.3. Policy Selection

All results are reported over a fixed sample of 30 policies, selected once and reused across every system, pool condition, and metric. For selecting the 30 policies, we draw a *tercile-stratified* sample: the 71-policy graph is sorted by argument count and split into three equal bins (sparse, medium, dense), and 10 policies are drawn uniformly at random from each (fixed seed, for reproducibility). The evaluation sample therefore spans the corpus's full density range by construction rather than by chance, so any relationship we observe between graph density and retrieval quality is not an artifact of oversampling one density regime.

4.4. Hardened Pools and Distractor Injection

The natural retrieval pool for a policy—its own set of arguments in a curated corpus—is largely relevant by construction: most of these arguments genuinely bear on the policy, since they were authored for it. Cuconasu et al. [18] show that such easy pools understate the difficulty a retriever faces and can let weak retrievers appear competitive with strong ones. We therefore construct a *hardened* pool: for each target policy we inject distractor arguments from topically related but distinct sibling policies—preferring metapolicy co-membership and falling back to centroid-cosine similarity when no metapolicy match exists—so that the injected distractors are plausible near-misses rather than random noise.

Because a distractor's effect on a graph-based retriever depends not only on its presence but on how it is connected, we vary two things independently: whether distractors carry CONTRADICTS edges to the genuine arguments of the target policy, and whether they carry them to one another. This gives three configurations:

- **Isolated** — no edges at all. Distractors sit in the pool as disconnected nodes, reachable only if a system's candidate generation includes them directly. This tests whether a graph retriever's traversal excludes distractors simply because they are unconnected.
- **Leaf-attached** (no distractor–distractor edges) — a small minority of distractors connect directly to the target policy's genuine arguments (553 native–distractor edges out of 345K total contradicts edges in this scenario), but none connect to each other. Most distractors, lacking any edge at all, remain unreachable by graph traversal exactly as in the isolated-distractors scenario..
- **Clustered** (unrestricted) — distractors additionally retain CONTRADICTS edges among themselves (385K edges, of which 40K are distractor–distractor). a pair of mutually contradicting distractors forms a small, densely connected component that accrues PageRank jointly, so a distractor's centrality score reflects the whole region even though candidate generation still reaches only one hop from the native seed.

The clustered configuration inflates **PR-GraphRAG**'s distractor rate through this transplanted-pair effect, which we argue (Section 6.4) is an artifact of graph construction rather than a genuine robustness failure: no real sibling policy would contribute a distractor pair together with the edge between them. We therefore treat **leaf-attached** as the corrected configuration and report our main hardened-pool results under it (Section 5.1).

4.5. Ground Truth

Each (policy, stance) candidate pool is exhaustively graded on a four-point ordinal scale (0 irrelevant to 3 essential) by two LLM annotators from different model families (Llama 3.3:70b and gemma 4:31b), with a conservative MIN-rule consensus (the lower grade kept on disagreement). Grading the full pool, rather than only each system's top-k, avoids the incomplete-judgment bias that would penalize a system for retrieving legitimate arguments outside a fixed top-k ground truth. This follows evidence that LLM assessors can match trained human judges on graded relevance [20, 21], while the dual-annotator design bounds the inter-LLM disagreement documented by Faggioli et al. [22].

To score a retrieved argument, its text is matched back to a graded pool member. For every system but **PR-GraphRAG** this is an exact string match, since the retrieved text is a verbatim pool member. **PR-GraphRAG**'s one-hop expansion can return arguments from outside the candidate pool, which have no exact match. For these we fall back to the single nearest graded argument by embedding similarity, accepted only above a strict threshold (cosine ≥ 0.75) so a retrieved argument never inherits the grade of a different one that merely resembles it.

4.6. Metrics

UC1 (retrieval). We report relevance with **nDCG@k** [23, 24], our primary metric, using exponential gain ($2^r - 1$) so that placing a grade-3 argument above a grade-2 is rewarded far more than under linear gain — appropriate for a moderator who reads only the top of the list — and with **MeanRel@k**, the mean matched grade of the top-k (0–3 scale), a rank-insensitive and more directly interpretable companion. Under the hardened pool we additionally report **distractor rate@k**, the fraction of the top-k that originates from an injected distractor policy rather than the target policy, and **EILD** [25], a relevance-weighted intra-list diversity that, unlike raw diversity, is not inflated by retrieving off-topic material. We also report **conflict density**, the mean NLI contradiction probability over retrieved pro×con pairs, as a supplementary measure of whether a retrieved set contains genuine argumentative tension; we read it with caution for **PR-GraphRAG**, whose candidate pool is expanded using the same NLI model that computes it.

UC2 (summarization). We score the generated summary directly rather than the retrieved list, since UC2's unit of evaluation is what the generation stage produces. Retrieved arguments are passed to a summarizer (qwen3:8b) prompted for a short, stance-balanced summary, which we then evaluate on four automatic metrics, deliberately preferring these to a holistic LLM-as-judge rubric because each has a narrower, more verifiable failure mode. **Coverage** is the fraction of graded-relevant source arguments entailed by some summary sentence (sentence-level NLI, threshold 0.5), measuring omission. **Faithfulness** is the fraction of atomic claims extracted from the summary that are entailed by the retrieved sources, measuring unsupported assertion. **Stance balance** measures how far the summary's pro/con coverage departs from an even split. **BERTScore F1** [26] measures semantic similarity to an independent reference summary; to avoid rewarding stylistic self-similarity, the reference is produced by a model (gemma4:31b) distinct from both the summarizer and the faithfulness judge, and is regenerated per (pool condition, system) so that the comparison is always within a matched condition.

5. Results

5.1. Conflict-Map Retrieval (UC1)

Native pool. Table 1 reports nDCG@k across $k \in \{5, 10, 15, 20, 30\}$, averaged over 30 stratified policies, under the native candidate pool (pool size 200, arguments drawn only from the target policy).

Under the native pool the four systems cluster within a narrow band — 0.32 to 0.40 at k=5 — and no system is clearly separated from the others. **RAG-no-MMR** is marginally highest at every k, ahead of **StandardRAG** by 3–4% relative, but **Random**, which uses no relevance signal at all, is within 0.006

Table 1
nDCG@k (exponential gain, mean of pro and con stances), native pool, averaged over 30 policies. Higher is better. **Random**'s competitiveness reflects that it draws from the same policy-restricted pool as every other system, and is a symptom of pool saturation rather than of retrieval quality (Section 6).

System	k=5	k=10	k=15	k=20	k=30
StandardRAG	0.390	0.390	0.397	0.405	0.422
PR-GraphRAG	0.324	0.335	0.343	0.351	0.369
RAG-no-MMR	0.404	0.405	0.412	0.421	0.439
Random	0.384	0.400	0.404	0.411	0.440

of **StandardRAG** throughout. **PR-GraphRAG** sits slightly below the other three (roughly 0.06–0.07 behind **StandardRAG**). That a signal-free baseline keeps pace with the engineered systems is the central feature of this table, not an incidental one: since we use a curated corpus, the native pool is not hard enough to distinguish the systems apart, and Section 6 reads it as motivation for the hardened protocol rather than as evidence about retrieval quality.

Table 2
MeanRel@k (mean matched grade, 0–3 scale), EILD, and conflict density, native pool, k=5, averaged over 30 policies.

System	MeanRel@k	EILD	Conflict density
StandardRAG	1.667	0.406	0.395
PR-GraphRAG	1.490	0.245	0.883
RAG-no-MMR	1.730	0.160	0.728
Random	1.637	0.416	0.446

MeanRel@k (Table 2) reproduces the same narrow ordering, confirming the near-parity is not a ranking artifact but reflects genuinely similar retrieved grades. The one clear separation in this table is on diversity: **StandardRAG**'s EILD (0.406) is more than double **RAG-no-MMR**'s (0.160), for a 3–4% relevance cost. Under easy conditions, in other words, MMR buys a large diversity gain cheaply — a trade Section 6 shows does not survive the hardened pool. Conflict density is higher for **PR-GraphRAG** and **RAG-no-MMR** (0.883, 0.728) than for **StandardRAG** and **Random** (0.395, 0.446), though we read **PR-GraphRAG**'s value with the caution noted in Section 4.6.

Hardened pool. Table 3 reports the same metrics under the hardened leaf-attached pool (Section 4.4), where roughly a third of each candidate pool is injected sibling-policy distractors.

Table 3
UC1 results under the hardened (leaf-attached) pool, k=5, pool= 200, averaged over 30 policies. Distractor rate: lower is better; all others higher is better.

System	nDCG@k	MeanRel@k	Distractor rate@k	EILD	Conflict density
StandardRAG	0.220	0.883	0.450	0.416	0.334
PR-GraphRAG	0.318	1.470	0.000	0.262	0.871
RAG-no-MMR	0.376	1.640	0.000	0.147	0.782
Random	0.246	1.090	0.323	0.354	0.437

The hardening separates the systems that the native pool could not. **PR-GraphRAG** and **RAG-no-MMR** both retrieve no distractors at all (rate 0.000), compared to 0.450 for **StandardRAG** and 0.323 for **Random**. This reverses the native-pool ordering on relevance: **PR-GraphRAG** now leads **StandardRAG** on both nDCG (0.318 vs. 0.220) and MeanRel@k (1.470 vs. 0.883), since every distractor **StandardRAG** retrieves is a wasted slot. **RAG-no-MMR** is the highest of all on both relevance metrics

— but at the lowest diversity of any system (EILD 0.147), whereas **PR-GraphRAG** reaches the same distractor immunity while retaining substantially more diversity (EILD 0.262). Section 6 interprets this pattern, and in particular what the **RAG-no-MMR** ablation reveals about the cause of **StandardRAG** distractor vulnerability.

5.2. Debate Summarization (UC2)

We evaluate UC2 at k=15 (Section 4.2), retrieving each system's top-15 pro and con arguments per policy, passing them to the summarizer, and scoring the resulting summary on the four metrics of Section 4.6. Table 4 reports the native pool and Table 5 the hardened leaf-attached pool, testing whether UC1's retrieval-time distractor resistance propagates to the generation stage.

Table 4
UC2 results, native pool, k=15, averaged over 30 policies.

System	Coverage	Faithfulness	Stance balance	BERTScore F1
StandardRAG	0.030	0.764	0.100	0.687
PR-GraphRAG	0.012	0.747	0.000	0.707
RAG-no-MMR	0.040	0.740	0.089	0.687
Random	0.046	0.776	0.160	0.686

Table 5
UC2 results, hardened (leaf-attached) pool, k=15, averaged over 30 policies.

System	Coverage	Faithfulness	Stance balance	BERTScore F1
StandardRAG	0.060	0.832	0.260	0.700
PR-GraphRAG	0.060	0.834	0.170	0.720
RAG-no-MMR	0.110	0.815	0.260	0.700
Random	0.050	0.715	0.190	0.690

The UC2 picture is far less separated than UC1's. Coverage is uniformly low under both pools (0.012–0.11), and BERTScore F1 varies by at most 0.03 across all systems in either condition. Under the native pool, **PR-GraphRAG**'s stance balance is a notable outlier at 0.000 — its summaries are effectively one-sided — and **Random** leads on both coverage and stance balance. Under the hardened pool, **RAG-no-MMR** leads coverage (0.11, nearly double the 0.06 of **StandardRAG** and **PR-GraphRAG**); **PR-GraphRAG** and **StandardRAG** tie on faithfulness (0.834 vs. 0.832) and **PR-GraphRAG** leads BERTScore F1 (0.72) but trails on stance balance (0.17 vs. 0.26). **PR-GraphRAG**'s clean retrieval in UC1 thus does not carry over to a clean win on summary quality; Section 6 argues that retrieval-stage robustness and generation-stage quality are only loosely coupled.

6. Discussion

6.1. The native-pool relevance–diversity trade

The native-pool comparison between **StandardRAG** and **RAG-no-MMR** (Table 2) sets a baseline for what follows. **StandardRAG** gives up 3–4% relevance relative to the MMR-free **RAG-no-MMR** in exchange for more than double the diversity (EILD 0.406 vs. 0.160). **PR-GraphRAG** does not offer this trade on a clean pool: it trails **StandardRAG** on both relevance and diversity (Section 5.1), so under easy conditions plain embedding similarity with MMR looks like the best choice.

The hardened pool suggests this trade is contingent on the pool being clean rather than a fixed property of MMR. **StandardRAG**'s version does not survive distractor pressure: **RAG-no-MMR** overtakes it on both relevance and distractor rate (Table 3). **PR-GraphRAG** appears to be the only

system that keeps a favorable trade under these conditions — distractor-immune like **RAG-no-MMR**, but more diverse, at a relevance cost of roughly the same order **StandardRAG** pays on the native pool. It is not the most relevant system, nor the most diverse, but the one that keeps the trade worth making once the pool can no longer be assumed clean.

6.2. Graph structure is not necessary for distractor resistance

The hardened-pool results point to a more precise reading than "graph structure resists distractors." The **RAG-no-MMR** ablation reaches the same 0.000 distractor rate as **PR-GraphRAG** with no graph signal at all, so graph structure does not appear to be *necessary* for distractor resistance here.

A plausible mechanism is visible in the setup. **StandardRAG** and **RAG-no-MMR** both rank by cosine similarity to the pool *centroid*, which is dominated by the native two-thirds of the pool; distractors, coming from a distinct sibling policy, sit systematically further from it, so plain top-k under that signal does not reach them within k=5. It is only once MMR's dissimilarity term begins rewarding distance from the already-selected set that the same pool, under the same relevance signal, surfaces them at a rate of 0.450.

6.3. What graph structure buys is safe diversification

MMR's dissimilarity term treats "distinct from what has already been selected" as evidence of useful diversity. This is reasonable on a clean pool, but becomes a liability once the pool contains a cluster that is distinct *because* it is off-topic. **PR-GraphRAG**'s candidate generation excludes that cluster before MMR runs, so diversification is applied only to genuine material.

This is why **RAG-no-MMR**'s nDCG lead over **PR-GraphRAG** (0.376 vs. 0.318) does not on its own settle the comparison. Diversity is part of what the conflict-map task asks for, not a quantity to be traded away freely: a set of five near-duplicate objections has arguably failed the task regardless of its grade. **RAG-no-MMR** reaches its distractor immunity by dropping diversification altogether, and its EILD (0.147) is the lowest of any system in Table 3. **PR-GraphRAG** reaches the same 0.000 rate while retaining more diversity (EILD 0.262) at a 15% relevance cost. We do not claim graph structure is the most diverse option — **StandardRAG** scores higher (0.416) — only that it is the one configuration we tested that is distractor-immune *and* diversified.

That last comparison carries a caveat. EILD averages over items graded ≥ 2, so **StandardRAG**'s 0.450 distractor rate leaves it averaging over only two or three surviving items against five for **PR-GraphRAG**, and mean pairwise distance over a two-item set is not directly comparable to one over five. **StandardRAG**'s apparent diversity edge should be read with that in mind; the **PR-GraphRAG**-versus-**RAG-no-MMR** comparison, where both retrieve five relevant items, is not subject to it.

On the native pool this same pre-filtering is a net cost, excluding graph-unreachable but topically relevant arguments without excluding anything MMR would have mishandled, since there is no adversarial cluster there to be misled by.

6.4. Robustness depends on distractor topology, not on distractor presence

Because this protection is structural, it holds only under particular assumptions about how distractors sit in the graph — assumptions our three configurations make explicit, and whose contrast is itself a result. Under **isolated** and **leaf-attached** distractors, **PR-GraphRAG** is unreachable-by-construction and its distractor rate stays near zero. Under the clustered configuration — distractors retaining CONTRADICTS edges to one another — the protection disappears: a pair of mutually contradicting distractors forms a small, densely connected component that accrues PageRank jointly, so multi-hop expansion that reaches either member reaches the whole region. **PR-GraphRAG**'s distractor rate rises to roughly 0.44, close to **StandardRAG**'s, and its nDCG advantage goes with it.

We treat leaf-attached as the corrected measurement: transplanting a distractor pair together with the edge between them injects structure no real sibling policy would contribute. But the contrast matters in its own right — leaf-attached distractors are dead ends for traversal, while distractors forming their own

connected region are a walkable neighborhood a centrality-seeded traversal will enter. The robustness of Section 5.1 is thus a property of the distractor subgraph's *topology*, not of "using the graph" as such, which suggests that an adversarial pool protocol for graph-RAG should specify how distractors are wired, not merely that they are present — the two choices can lead to opposite conclusions about the same system.

6.5. Pool difficulty as an evaluation variable

More generally, reporting only native-pool results risks concluding that graph structure adds nothing, when the more accurate statement is that it adds nothing *under easy conditions that may not reflect deployment*. A participatory-democracy platform in production is closer to the adversarial case: competing proposals on related policies are the kind of topically similar, policy-distinct distractor our hardened pool is meant to simulate. **Random**'s competitive native-pool nDCG (Table 1) is itself evidence for this — a method with no relevance signal keeps pace with a graph-aware system simply because the native pool cannot tell them apart — and the hardened protocol restores that discriminative power. We therefore suggest treating pool difficulty as an explicit evaluation variable rather than an implementation detail.

6.6. Retrieval-stage robustness does not clearly propagate downstream

The UC2 results qualify the picture above. **PR-GraphRAG**'s clean retrieval does not translate into a correspondingly clean summary: it leads on BERTScore F1 and ties on Faithfulness, but trails on Coverage and Stance balance, and all four systems separate much less on summarization metrics than on retrieval metrics. Retrieval-time distractor resistance and summary quality appear only loosely coupled, with the summarizer's own behavior intervening between a clean candidate set and a good summary. We report this as a negative result: on this corpus, retrieval-quality differences of the magnitude we measure seem to be largely absorbed by the generation stage.

6.7. Preliminary generalization to the Grand Débat National

To probe whether the mechanisms above are specific to the IBM-ArgQ-Rank-30k corpus, we ran UC1 on a second corpus under the same thresholds, with no re-tuning. The Grand Débat National (GDN) is a French-language participatory corpus of citizen contributions to a 2019 national consultation, from which we drew 30 stratified policies following the same tercile procedure as Section 4.3, yielding 21,471 arguments and 4,921 CONTRADICTS edges under the identical blocking floor and NLI threshold calibrated on IBM. Unlike IBM-ArgQ-Rank-30k, GDN is a raw, unfiltered open-consultation text. A large share of it is not argumentative with respect to any given policy at all, 12,972 of the 21,471 arguments (60.4%) carry a neutral stance label and are excluded from every pro or con pool by construction, leaving a smaller and noisier argumentative core than IBM provides. Because GDN is both a different domain and a noisier, less curated corpus, its results should be read as a preliminary check rather than a general claim about French participatory debate

Against that backdrop, only 1,987 of the 21,471 arguments (9.3%) carry any CONTRADICTS edge at all, giving **PR-GraphRAG** a nonzero PageRank signal for a median of roughly 15 to 20 arguments per policy per stance before the remainder are tied at zero. Whether this shallow coverage reflects genuinely sparser argumentative structure in French participatory debate, the corpus's lack of curation, or both, cannot be disentangled from this experiment alone.

Table 6 reports nDCG@k on GDN's native pool. Unlike on IBM, **PR-GraphRAG** leads at k=5 and k=10, then falls behind all three other systems from k=15 onward, ending last at k=30.

Table 7 reports conflict density, the mean NLI contradiction probability over retrieved pro×con pairs, over the same grid. **PR-GraphRAG** leads at every k but decays steadily as k grows, while the other three systems stay flat.

The nDCG reversal in Table 6 tracks the decay in Table 7 closely, both consistent with the graph-coverage mechanism above: **PR-GraphRAG**'s advantage is concentrated in the arguments its PageRank

Table 6
nDCG@k (exponential gain, mean of pro and con stances), GDN native pool, averaged over 30 policies. Higher is better.

System	k=5	k=10	k=15	k=20	k=30
StandardRAG	0.337	0.364	0.391	0.410	0.466
PR-GraphRAG	0.356	0.366	0.385	0.406	0.449
RAG-no-MMR	0.338	0.377	0.419	0.448	0.496
Random	0.333	0.360	0.392	0.403	0.466

Table 7
Conflict density, GDN native pool, averaged over 30 policies. Higher is better.

System	k=5	k=10	k=15	k=20	k=30
StandardRAG	0.278	0.297	0.304	0.300	0.295
PR-GraphRAG	0.673	0.567	0.498	0.463	0.413
RAG-no-MMR	0.314	0.323	0.324	0.320	0.318
Random	0.305	0.307	0.316	0.293	0.301

signal reaches, and as k outruns that signal for a growing share of policies, both the relevance and the conflict-bearing content of its retrieved set are diluted by arguments selected with no real ranking signal.

Table 8 reports EILD over the same grid. Here the pattern breaks from the two tables above rather than reinforcing them. **PR-GraphRAG** starts lowest among the three MMR-using systems at k=5 (0.299) but overtakes all three by k=30 (0.355, the highest value in the table), the opposite trajectory from its nDCG and conflict-density decay. **RAG-no-MMR**, with no diversification mechanism, is lowest at every k, as expected.

Table 8
EILD, GDN native pool, averaged over 30 policies. Higher is better.

System	k=5	k=10	k=15	k=20	k=30
StandardRAG	0.329	0.338	0.350	0.351	0.349
PR-GraphRAG	0.299	0.312	0.348	0.351	0.355
RAG-no-MMR	0.207	0.250	0.275	0.279	0.292
Random	0.310	0.348	0.351	0.352	0.351

This decoupling is informative. Once k exceeds the available PageRank signal, **PR-GraphRAG**'s retrieved tail is effectively unranked, so its *relevance* degrades — but its *diversity* does not, since MMR's dissimilarity term keeps operating on that tail regardless of whether the relevance behind it is meaningful. What degrades at high k is specifically ranking, not diversification.

These results are consistent with the native-pool finding of Section 5.1: graph structure offers no fixed advantage, and any advantage is conditional on density. But GDN differs from IBM-ArgQ-Rank-30k in domain, language, and curation at once, and with 60.4% of its arguments labeled neutral it is far noisier by construction — so the shallow CONTRADICTS coverage we observe could reflect a real property of French participatory debate, the noise, or both. We treat this as a preliminary, single-corpus result; the hardened-pool analysis of Sections 6.4 and 6 remains untested on GDN.

6.8. Limitations

Several caveats bound these conclusions. All hardened-pool results come from the IBM-ArgQ-Rank-30k corpus; Section 6.7 tests the native-pool finding on a second corpus, but not the hardened-pool mechanism, which on noisier, user-generated deliberation remains untested. The ground-truth grades

are produced by a dual-annotator LLM consensus rather than by human annotators. LLM assessors have been shown to match trained judges on graded relevance [20, 21], which bounds this risk without removing it. The MMR failure mode is observed only at λ=0.5, and we have not swept λ to see whether it attenuates as diversification weight decreases. Finally, **PR-GraphRAG**'s one-hop expansion was fixed a priori rather than tuned, and deeper traversal would enlarge the candidate pool in a way that might erode the very structural filtering we identify as the source of its robustness.

7. Conclusion

We asked whether an argument graph, once recovered, improves retrieval for argumentative conflict maps, and found the answer depends on how the question is posed. On a native pool—each policy's own arguments, mostly relevant by construction—graph-aware retrieval offers no advantage over an embedding-only baseline; the pool is easy enough that even random selection is competitive, so any conclusion drawn from it says more about the evaluation than the systems. Under an adversarially hardened pool seeded with sibling-policy distractors the picture changes, but not as one might assume: removing diversification entirely, with no graph signal at all, reaches the same distractor immunity as our graph-aware system, so graph structure is not what confers robustness.

What the graph confers is narrower and, we think, more interesting. The vulnerability lies in the diversification step, whose dissimilarity term cannot tell an argument that is distinct because it is novel from one that is distinct because it is off-topic, and under distractor pressure is drawn toward the injected arguments. Generating candidates by graph traversal removes those arguments before diversification runs, letting a system stay distractor-resistant without giving up diversity—the trade **StandardRAG** loses to this failure mode and that **RAG-no-MMR** avoids only by dropping diversification. But this is not a general property of using a graph: it holds only for distractors that attach to the genuine graph as dead ends; when they form their own connected region, a centrality-seeded traversal walks into it and the advantage disappears.

The methodological point is that pool difficulty and distractor wiring are not details to be fixed at a convenient default. They determine which system appears to win: the four systems are indistinguishable on an easy pool, separate once distractors are injected, and reverse again depending on how those distractors are connected. Graph-RAG evaluations that report a single easy pool, or inject distractors without specifying their topology, risk conclusions a different but equally reasonable pool would overturn. We therefore suggest treating both as explicit, reported variables.

Two directions follow from the limitations above. Section 6.7 takes a first step on the Grand Débat National, a French-language participatory corpus, but only in the native-pool setting; extending the hardened-pool analysis there would test whether the topology-dependent mechanism of Section 6.4 survives noisier, user-generated argumentation. Second, a human-labeled sample of the ground truth would gauge how far the LLM annotations diverge from human judgment, which the dual-annotator consensus bounds only indirectly.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Gemini, Claude and Grammarly in order to rephrase some of the sentences and also to fix grammar and spelling issues. After using these tools and services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

8. Acknowledgements

This work is supported under the France 2030 program by the ANR-23-IACL-0007 grant (AI Cluster PostGenIA).

References

[1] X. E. Barandiaran, A. Calleja-López, A. Monterde, C. Romero, Decidim, a Technopolitical Network for Participatory Democracy, Springer, 2024. Open access.

[2] B. Shin, J. Floch, M. Rask, P. Bæck, C. Edgar, A. Berditchevskaia, P. Mesure, M. Branlat, A systematic analysis of digital tools for citizen participation, Government Information Quarterly 41 (2024) 101954. doi:10.1016/j.giq.2024.101954.

[3] M. Arana-Catania, F.-A. V. Lier, R. Procter, N. Tkachenko, Y. He, A. Zubiaga, M. Liakata, Citizen participation and machine learning for a better democracy, Digital Government: Research and Practice 2 (2021) 1–22. URL: http://dx.doi.org/10.1145/3452118. doi:10.1145/3452118.

[4] W. Aboucaya, O. Balalau, R. Angarita, V. Issarny, Finding conflicts of opinion in citizen participation platforms, in: 2024 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2024, pp. 70–77. doi:10.1109/WI-IAT62293.2024.00018.

[5] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, J. Larson, From local to global: A graph RAG approach to query-focused summarization, arXiv preprint arXiv:2404.16130 (2024).

[6] Z. Guo, L. Xia, Y. Yu, T. Ao, C. Huang, Lightrag: Simple and fast retrieval-augmented generation, arXiv preprint arXiv:2410.05779 (2024).

[7] B. J. Gutiérrez, Y. Shu, Y. Gu, M. Yasunaga, Y. Su, Hipporag: Neurobiologically inspired long-term memory for large language models, Advances in neural information processing systems 37 (2024) 59532–59569.

[8] A. Bondarenko, M. Fröbe, J. Kiesel, S. Syed, T. Gurcke, M. Beloucif, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, M. Hagen, Overview of touché 2022: Argument retrieval, in: Working Notes of CLEF 2022 – Conference and Labs of the Evaluation Forum, volume 3180 of *CEUR Workshop Proceedings*, 2022, pp. 2867–2903.

[9] N. Falk, A. Waldis, I. Gurevych, Overview of PerspectiveArg2024: The first shared task on perspective argument retrieval, in: Proceedings of the 11th Workshop on Argument Mining (ArgMining 2024), 2024, pp. 130–149.

[10] S. Gretz, R. Friedman, E. Cohen-Karlik, A. Toledo, D. Lahav, R. Aharonov, N. Slonim, A large-scale dataset for argument quality ranking: Construction and analysis, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 7805–7813.

[11] O. Cocarascu, F. Toni, Identifying attack and support argumentative relations using deep learning, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2017, pp. 1374–1379.

[12] E. Cabrio, S. Villata, Combining textual entailment and argumentation theory for supporting online debates interactions, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012), 2012, pp. 208–212.

[13] M.-C. de Marneffe, A. N. Rafferty, C. D. Manning, Finding contradictions in text, in: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-08: HLT), 2008, pp. 1039–1047.

[14] J. Carbonell, J. Goldstein, The use of MMR, diversity-based reranking for reordering documents and producing summaries, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98), ACM, 1998, pp. 335–336.

[15] C.-N. Ziegler, S. M. McNee, J. A. Konstan, G. Lausen, Improving recommendation lists through topic diversification, in: Proceedings of the 14th International Conference on World Wide Web (WWW '05), 2005, pp. 22–32.

[16] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, I. MacKinnon, Novelty and diversity in information retrieval evaluation, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08), 2008, pp. 659–666.

[17] Q. Lu, N. D. Sidiropoulos, Principled and scalable diversity-aware retrieval via cardinality-constrained binary quadratic programming, arXiv preprint arXiv:2604.02554 (2026).

[18] F. Cuconasu, G. Trappolini, F. Siciliano, S. Filice, C. Campagnano, Y. Maarek, N. Tonellotto,

F. Silvestri, The power of noise: Redefining retrieval for rag systems, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, ACM, 2024, p. 719–729. URL: http://dx.doi.org/10.1145/3626772.3657834. doi:10.1145/3626772.3657834.

[19] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web, Technical Report 1999-66, Stanford InfoLab, 1999.

[20] P. Thomas, S. Spielman, N. Craswell, B. Mitra, Large language models can accurately predict searcher preferences, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), 2024, pp. 1930–1940.

[21] S. Upadhyay, R. Pradeep, N. Thakur, N. Craswell, J. Lin, UMBRELA: UMbrela is the (open-source reproduction of the) Bing RELevance Assessor, arXiv preprint arXiv:2406.06519 (2024).

[22] G. Faggioli, L. Dietz, C. L. A. Clarke, G. Demartini, M. Hagen, C. Hauff, N. Kando, E. Kanoulas, M. Potthast, B. Stein, H. Wachsmuth, Perspectives on large language models for relevance judgment, in: Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '23), 2023, pp. 39–50.

[23] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Transactions on Information Systems 20 (2002) 422–446.

[24] N. Thakur, L. Bonifacio, M. Fröbe, A. Bondarenko, E. Kamalloo, M. Potthast, M. Hagen, J. Lin, Systematic evaluation of neural retrieval models on the touché 2020 argument retrieval subset of beir, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 1420–1430. URL: https://doi.org/10.1145/3626772.3657861. doi:10.1145/3626772.3657861.

[25] D. Carraro, D. Bridge, Enhancing recommendation diversity by re-ranking with large language models, 2024. URL: https://arxiv.org/abs/2401.11506. arXiv:2401.11506.

[26] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: Evaluating text generation with BERT, in: Proceedings of the 8th International Conference on Learning Representations (ICLR 2020), 2020.

9. Online Resources

- GitHub