

Support-Aware Retrieval of Evidence Passages for Community Notes

Stanford CS224N Custom Project

Dorian Gulley

Department of Computer Science
Stanford University
dgulley@stanford.edu

Dyllan Han

Symbolic Systems
Stanford University
dyllanh@stanford.edu

Abstract

X’s Community Notes is a widely used crowdsourced fact-checking system that attaches short, source-backed notes to potentially misleading posts. Although its ranking mechanism is designed to surface notes judged helpful across viewpoints, the quality of cited evidence remains uneven in practice. Sources may be outdated, only loosely related to the underlying claim, or cumbersome for readers to verify, which can weaken the persuasive force of otherwise accurate notes. This makes evidence retrieval difficult: useful evidence must be both relevant to the note and grounded in accessible, verifiable sources.

In this project, we study a retrieval-augmented evidence pipeline for Community Notes centered on Wikipedia. We construct a targeted corpus from Wikipedia articles cited in helpful Community Notes. We then compare two hybrid retrieval pipelines: a Hybrid system that combines BM25 and dense retrieval with Reciprocal Rank Fusion, and a Hybrid+Rerank variant that adds a reranking stage. We evaluate these methods against BM25 and Gemini-based baselines using source-level Recall@ k and an NLI-based support metric.

Our main finding is that hybrid retrieval improves recovery of gold Wikipedia sources, but does not produce large gains in passage-level semantic support. Hybrid and Hybrid+Rerank both outperform BM25 on Recall@5, with Hybrid+Rerank achieving the best overall score, while BM25 remains competitive on Support-Score@5. Gemini performs substantially worse under strict source-level evaluation. These results suggest that hybrid retrieval is a promising strategy for improving citation recovery in Community Notes, while further gains may depend more on better passage selection than on reranking alone.

1 Key Information to include

- Mentor: Nevin George
- External Collaborators (if you have any): No
- Sharing project: No
- Late Days (if you are using any): 1

Team member	Late days remaining
Dorian Gulley	1
Dyllan Han	2
Total pooled	3
Team size	2
Late days applied for the report	1

2 Introduction

X’s Community Notes is a crowdsourced fact-checking system that allows contributors to attach short, source-backed notes to posts that may be misleading. A key feature of the system is its ranking mechanism, which promotes notes judged helpful by users with differing viewpoints. This design has made Community Notes a compelling example of large-scale, community-driven moderation: rather than relying only on expert review, it attempts to surface corrections that appear broadly credible across ideological groups.

However, the usefulness of a note depends not only on how it is written, but also on the quality of the evidence it cites. In practice, supporting sources are often uneven. Some are outdated, some are only tangentially related to the claim, and some are difficult for readers to inspect quickly. These limitations matter because Community Notes must persuade under strong attention constraints. Even when a note is accurate, weak or hard-to-verify evidence can reduce its credibility and limit its practical impact.

Retrieving good evidence for Community Notes is therefore a challenging problem. Strong supporting evidence must satisfy several criteria at once: it should be relevant to the note, specific enough to support the claim being made, and drawn from a source that readers can verify with little effort. Existing retrieval methods only partially address this challenge. Lexical retrieval methods such as BM25 are often effective when the note and source share terminology, but they can miss useful evidence when relevant passages are phrased differently. Dense retrieval can recover semantically related content beyond exact word overlap, but it may also retrieve passages that are broadly related without being the best citation for the claim. Generative systems can sometimes propose plausible sources, yet under strict source-level evaluation they may be inconsistent and are more vulnerable to hallucinated evidence.

In this project, we investigate whether a retrieval-augmented pipeline can improve evidence retrieval for Community Notes by focusing on Wikipedia as a high-coverage, relatively accessible source of supporting information. We construct a targeted corpus from Wikipedia articles cited in helpful notes. We then study two hybrid retrieval pipelines: a Hybrid approach that combines BM25 and dense retrieval using Reciprocal Rank Fusion, and a Hybrid+Rerank variant that adds a reranking stage. We evaluate these methods against BM25 and Gemini-based baselines using source-level Recall@ k and an NLI-based support metric.

3 Related Work

Our project sits at the intersection of crowdsourced fact-checking, retrieval-augmented generation (RAG), and evaluation of evidence quality. Prior work in these areas motivates both our problem formulation and our design choices.

Community Notes and evidence quality. X’s Community Notes is a crowdsourced system for attaching source-backed notes to potentially misleading posts. A key feature of the system is that note visibility depends on broad agreement across users with differing viewpoints, which is intended to make surfaced notes more credible and more resistant to manipulation than simple majority-vote systems [1, 2]. At the same time, the official Community Notes documentation emphasizes that building such a system remains challenging in practice, including ensuring resistance to gaming and helping contributors identify sources that readers will find trustworthy and useful [1]. Our work focuses on this latter issue: even when a note is rated helpful, the quality and accessibility of its supporting evidence may still vary. This makes evidence retrieval an important component of improving the persuasiveness and verifiability of Community Notes.

Retrieval for knowledge-grounded generation. Retrieval-augmented generation has become a standard approach for grounding language models in external knowledge. Recent work has shown that RAG can improve factuality and adapt models to knowledge-intensive tasks, but it also depends heavily on retrieving the right evidence in the first place [3, 4]. RankRAG, for example, highlights a central tradeoff in RAG pipelines: retrieving too few contexts hurts recall, while retrieving too many can introduce irrelevant or noisy evidence that harms downstream performance [3]. This observation is directly relevant to our setting, where Community Note summaries are short and evidence quality matters more than sheer context volume. Motivated by this tradeoff, we focus on the

retrieval stage itself and compare lexical, dense, and fused retrieval methods for recovering useful supporting passages from a targeted Wikipedia corpus.

Robustness to noisy or misleading evidence. A growing line of work argues that standard RAG evaluation often assumes unrealistically clean retrieval. RAGUARD shows that when models are exposed to naturally misleading retrieved evidence, performance can degrade substantially, in some cases falling below zero-shot baselines [5]. This is especially relevant for fact-checking applications, where retrieved evidence may be partial, noisy, or framed in ways that subtly mislead a model. Our project addresses a narrower but related problem: rather than studying robustness to arbitrary retrieved evidence, we ask whether evidence retrieval for Community Notes can be improved by centering retrieval on Wikipedia, a source family already used in many helpful notes and generally easier to verify quickly. In this sense, our work complements robustness-oriented RAG research by focusing on corpus design and retrieval quality as a first step toward more reliable evidence grounding.

Evaluation of retrieval quality and support. Evaluating RAG systems is challenging because retrieval quality and generation quality are intertwined. RAGCHECKER argues for fine-grained evaluation that separates retriever behavior from generator behavior and uses claim-level entailment to diagnose whether retrieved context actually supports the final output [4]. We adopt a similar intuition in our evaluation design. However, because our project focuses on the first stage of the pipeline, we evaluate primarily at the retrieval level: source-level Recall@ k measures whether gold cited sources are recovered, while an NLI-based support score estimates whether retrieved passages semantically support the note summary. This lets us assess retrieval quality even before full note generation.

NLI as a support signal. Our support metric is also motivated by prior work on natural language inference as a general-purpose mechanism for judging semantic compatibility between texts. Laurer and colleagues show that NLI can serve as a flexible universal classification framework and support a wide range of zero-shot classification tasks [6]. In our setting, we do not use NLI as the primary retrieval mechanism, but rather as a lightweight way to estimate whether a retrieved passage supports the content of a Community Note summary. This makes NLI a useful complement to source-level recall: a system may fail to recover the exact cited URL while still retrieving passages that provide strong semantic support, or conversely may recover a cited source without retrieving a clearly supportive passage.

Positioning our work. Taken together, these strands of prior work suggest that effective evidence support for Community Notes depends on more than simply attaching any citation. Community Notes highlights the importance of broad trust and verifiability; RAG research highlights the importance of high-quality retrieval; robustness work highlights the risk of misleading or irrelevant evidence; and NLI-based evaluation offers a way to measure semantic support. Our project builds on these ideas by studying a targeted Wikipedia retrieval pipeline for Community Notes, with two main contributions: an original citation-derived corpus design centered on Wikipedia articles cited in helpful Community Notes, and hybrid retrieval approaches that combine BM25, dense retrieval, and reranking to improve evidence recovery.

4 Approach

We study the first stage of an evidence retrieval pipeline for Community Notes. The input to the system is a Community Note, represented by its natural-language summary, and the output is a ranked list of supporting evidence excerpts paired with their source URLs:

$$(\text{Community Note}) \rightarrow [(\text{source}_1, \text{excerpt}_1), (\text{source}_2, \text{excerpt}_2), \dots, (\text{source}_k, \text{excerpt}_k)].$$

Our goal is to retrieve evidence that is both relevant to the note content and grounded in verifiable Wikipedia sources, thereby improving the quality and persuasiveness of supporting citations.

We use the note summary as the retrieval query after removing URLs to avoid trivial citation leakage. Candidate evidence is retrieved at the passage level so that the system returns directly usable supporting excerpts rather than only article titles or document identifiers.

Our approach combines two complementary retrieval signals. First, we apply BM25 using Pyserini’s Lucene index [7]. For query q and passage p , BM25 assigns the score

$$\text{score}_{\text{BM25}}(q, p) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{f(t, p) \cdot (k_1 + 1)}{f(t, p) + k_1 \left(1 - b + b \frac{|p|}{\text{avgdl}}\right)}.$$

BM25 captures direct lexical overlap between the note summary and candidate evidence passages.

Second, we apply dense retrieval using passage and query embeddings with nearest-neighbor search. Given encoder $f(\cdot)$, the query and passage are represented as dense vectors $\mathbf{e}_q = f(q)$ and $\mathbf{e}_p = f(p)$, and ranked by inner-product similarity:

$$\text{score}_{\text{dense}}(q, p) = \mathbf{e}_q^\top \mathbf{e}_p.$$

Dense retrieval is intended to recover semantically relevant passages even when there is limited term overlap with the note summary.

Our main methodological contribution is a hybrid retrieval framework that combines lexical and semantic retrieval while also allowing us to test the value of reranking. In the **Hybrid** pipeline, we retrieve passages using both BM25 and dense retrieval, then combine the two rankings with Reciprocal Rank Fusion (RRF) [8]:

$$\text{RRF}(p) = \sum_{s \in \{\text{BM25}, \text{dense}\}} \frac{1}{c + \text{rank}_s(p)},$$

where $\text{rank}_s(p)$ denotes the rank of passage p under system s , and c is the fusion constant. This fusion step rewards passages that rank well under either retrieval signal while favoring those that are supported by both.

To test whether a second-stage ranking model improves over fusion alone, we also consider a **Hybrid+Rerank** pipeline. This variant begins from the same BM25+dense candidate pool and RRF fusion stage, then applies an additional reranking step to reorder the fused candidates. This design lets us distinguish the benefit of hybrid retrieval itself from the benefit of a more refined second-stage ranking model.

For evaluation, retrieved passages are mapped back to their corresponding `source_url` values so that we can measure whether the system recovers the gold Wikipedia citations associated with each note. We also assess whether retrieved passages provide semantic support for the note content using an entailment-style metric described in the Experiments section.

In addition to these retrieval-based systems, we include a generation-based baseline using Gemini. Rather than retrieving from the indexed corpus, Gemini is asked to propose a supporting Wikipedia source directly from the note summary. We use this baseline to test whether a strong parametric model can recover gold citations without explicit corpus retrieval.

5 Experiments

5.1 Data

Our experiments use two Community Notes tables downloaded from X: a notes table (about 1.2GB) and a ratings table (about 0.7GB), both current through mid-February 2026. We derive a reduced notes dataset by selecting notes that are currently rated *helpful* and include at least one Wikipedia citation. From each retained note, we keep only `noteId`, `tweetID`, `summary`, `classification`, and `trustworthySources`. This filtering yields 14,624 notes in a 6MB file. We then filtered these to English-only notes, which yielded 3,291, and then randomly sampled 1,000 community notes from this.

We construct the retrieval corpus from the Wikipedia articles cited by these notes. Article URLs are normalized and deduplicated, and article text is fetched through the MediaWiki API. The resulting articles are converted into retrieval passages by splitting plaintext into overlapping chunks of roughly 200 words with 50-word overlap, using sentence boundaries when possible. Very short pages are excluded, and both URLs and passage texts are deduplicated before the corpus is serialized as JSONL for Pyserini indexing. Each passage stores its text and `source_url` metadata for later source-level evaluation.

5.2 Evaluation method

We evaluate retrieval quality primarily at the source level using the gold citations provided in each note’s `trustworthySources` field. For a query q , let $G(q)$ denote its set of gold sources, defined as the set of Wikipedia URLs listed in `trustworthySources` after URL normalization and deduplication. Given a ranked list of retrieved passages, we construct $R_k(q)$ by taking the first k *unique* `source_url` values in rank order. We then compute source-level $\text{Recall}@k$ as

$$\text{Recall}@k = \frac{1}{|Q|} \sum_{q \in Q} \mathbf{1}(G(q) \cap R_k(q) \neq \emptyset),$$

which measures whether at least one gold citation is recovered among the top- k retrieved sources for each query.

In addition to citation recall, we evaluate whether retrieved passages substantively support the note content. For this, we score each query–passage pair with the pretrained NLI model `MoritzLaurer/DeBERTa-v3-base-mnli`. For each retrieved passage, the model produces an entailment-style support score, and we define $\text{SupportScore}@k$ as the average of these scores over the top- k retrieved passages for each query, followed by averaging across queries. This metric captures whether the retrieved evidence is semantically supportive of the note, even when the exact cited source is not retrieved.

5.3 Experimental details

All experiments are run in inference-only mode; we do not perform any task-specific training or fine-tuning. The evaluation cutoff is fixed at $k = 5$, and each retrieval system returns a ranked list of the top $N = 10$ passages before source-level deduplication.

For lexical retrieval, we use Pyserini with a Lucene BM25 index over the passage corpus [7]. For dense retrieval, we encode queries and passages with the pretrained sentence-transformer `all-MiniLM-L6-v2`. Embeddings are L2-normalized, and nearest-neighbor search is performed with FAISS `IndexFlatIP`. Both BM25 and dense retrieval return the top 10 passages per query.

For the Hybrid and Hybrid+Rerank pipelines, we combine BM25 and dense rankings using Reciprocal Rank Fusion. The Hybrid system uses the fused ranking directly, while Hybrid+Rerank reranks the fused passage candidates with the pretrained cross-encoder `cross-encoder/ms-marco-MiniLM-L-6-v2`, which scores each (query, passage) pair directly. The top 10 reranked passages per note are retained for evaluation.

For support-based evaluation, we score retrieved query–passage pairs with `MoritzLaurer/DeBERTa-v3-base-mnli`. NLI inference is run in batches of 32 passages.

Gemini 2.0 Flash is evaluated separately as a recall-only baseline. To avoid rate-limiting constraints during API-based evaluation, we run Gemini on a random 500-note subset rather than the full 1,000-note evaluation set. We prompt Gemini to return a list of Wikipedia URLs that best support a Community Note summary:

```
Given this Community Note summary, list the {k} most relevant
Wikipedia article URLs that support or verify the claims made.
Return only a JSON array of Wikipedia URLs in order of relevance.
Example: ["https://en.wikipedia.org/wiki/X", ...]
```

```
Community Note: {query}
```

Each returned URL is normalized to a Wikipedia URL before computing source-level recall.

Because retrieval, fusion, reranking, and NLI scoring are deterministic given fixed inputs and model versions, we report single-run results without error bars.

Our experiments therefore compare four systems under a common setup: BM25, Hybrid, Hybrid+Rerank, and Gemini as a generation-based recall baseline.

System	Notes	Recall@5	SupportScore@5
BM25	1000	0.8151	0.6208
Hybrid (BM25 + dense RRF)	1000	0.8647	0.6199
Hybrid + Rerank	1000	0.8736	0.6210
Gemini 2.0 Flash (parametric)	500	0.5217	N/A

Table 1: Main results on source-level recall and NLI-based support. Higher is better for both Recall@5 and SupportScore@5. Gemini is evaluated only with recall, since its free-form generations may produce hallucinated or unverifiable evidence.

5.4 Results

Results. Table 1 reports the main quantitative results. Following our evaluation setup, we compare BM25, hybrid retrieval with BM25+dense fusion, hybrid retrieval with an additional reranking stage, and Gemini 2.0 Flash as a parametric baseline. We fix the evaluation cutoff at $k = 5$.

Overall, the hybrid+rerank retrieval system achieves the strongest performance among the retrieval-based methods. Relative to BM25 alone, combining lexical and dense retrieval through RRF improves Recall@5 from 0.8151 to 0.8647, indicating that the hybrid system is better able to recover gold supporting sources. The hybrid and rerank system yields the highest SupportScore@5, though the gain over BM25 is very small (0.6210 vs. 0.6208). This suggests that lexical retrieval already finds semantically relevant passages in many cases, but that fusion with dense retrieval improves source recovery without sacrificing passage-level support.

Adding the reranking stage provides only marginal improvements over the hybrid baseline. Hybrid+Rerank achieves 0.8736 Recall@5 and 0.6210 SupportScore@5, representing minimal gains over the simpler hybrid system (0.8647 and 0.6199 respectively). This modest performance suggests that while reranking can modestly refine passage order, the gains may be offset by disruptions to source diversity in the top-ranked results. Since our primary metric is source-level recall, preserving a broader mix of candidate sources appears more useful than pursuing local reranking.

Gemini performs substantially worse than all retrieval-based methods on recall, with Recall@5 of 0.5217. This result is consistent with our motivation for treating Gemini only as a recall baseline: although a parametric model can often generate plausible supporting evidence, it is much less reliable at recovering the exact gold Wikipedia sources used in Community Notes. In other words, fluent evidence generation does not necessarily translate into verifiable source recovery.

6 Analysis

Our results show a clear difference between source recovery and semantic support. On Recall@5, the ranking is

$$\text{Hybrid} \approx \text{Hybrid+Rerank} > \text{BM25} \gg \text{Gemini},$$

showing that combining lexical and dense retrieval improves recovery of gold Wikipedia sources from Community Note summaries. BM25 alone is already strong, but adding dense retrieval appears to improve source coverage by capturing semantically related passages beyond exact term overlap. Gemini performs much worse under strict source-level recall, which supports our decision to use it only as a recall baseline rather than as a full retrieval system.

The pattern is weaker for SupportScore@5, where

$$\text{Hybrid} \approx \text{Hybrid+Rerank} \approx \text{BM25}.$$

This suggests that while hybrid retrieval improves *which sources* are recovered, it does not substantially improve the average semantic support of the top retrieved passages. One likely reason is that BM25 already retrieves passages that are reasonably supportive even when it misses the exact gold source.

We also find that reranking does not significantly improve the hybrid pipeline. Hybrid+Rerank performs similarly to, but not substantially better than, the simpler Hybrid system on both recall and support. This suggests that reranking may refine local passage order without materially improving source diversity, which matters more for our source-level evaluation.

Overall, the main benefit of the hybrid approach is improved citation recovery rather than large gains in passage-level support. Future work should therefore focus not only on better retrieval coverage, but also on better passage selection for direct explanatory support.

7 Conclusion

In this project, we studied the evidence retrieval stage of a retrieval-augmented pipeline for X’s Community Notes. We constructed a targeted Wikipedia corpus from citations in helpful notes and compared lexical, dense, and hybrid retrieval methods for recovering supporting evidence from note summaries. Our main finding is that hybrid retrieval, combining BM25 and dense retrieval through Reciprocal Rank Fusion, improves source-level recall over the baselines. At the same time, gains on the NLI-based support metric are smaller, suggesting that better citation recovery does not automatically translate into substantially stronger passage-level support.

These results highlight both the promise and the limits of our approach. On the positive side, we show that a relatively simple hybrid retrieval pipeline can improve recovery of gold Wikipedia sources, which is an important step toward making Community Notes more verifiable and persuasive. We also find that reranking provides little additional benefit in our current setup, and that a parametric model such as Gemini is substantially less reliable under strict recall-based evaluation. More broadly, we learned that retrieval quality in this setting should be evaluated along multiple dimensions, since source recovery and semantic support capture different aspects of usefulness.

The main limitations of our work are that evaluation was conducted on a relatively small subset of notes, the corpus was restricted to Wikipedia-based evidence, and we focused only on retrieval rather than full note generation. Future work could scale evaluation to a larger sample, explore stronger reranking or passage selection methods, and study whether improved retrieval leads to more factual and persuasive generated notes.

Team contributions (Required for multi-person team)

Code for the project is available at https://github.com/dyllan-07/224N_Final.

Dorian Gulley:

- Preprocessed data with filter for rating and Wikipedia citations
- Constructed evaluation infrastructure
- Streamlined runs through a one-command pipeline

Dyllan Han:

- Constructed BM25 retrieval pipeline
- Constructed hybrid and hybrid+rerank retrieval modes
- Evaluated Gemini retrieval

References

- [1] Community Notes. Challenges. <https://communitynotes.x.com/guide/en/about/challenges>, 2026. Official Community Notes documentation. Accessed: 2026-03-11.
- [2] Community Notes. Note ranking algorithm. <https://communitynotes.x.com/guide/en/under-the-hood/ranking-notes>, 2026. Official Community Notes documentation. Accessed: 2026-03-11.
- [3] Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [4] Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang,

- Tong He, Zhiguo Wang, Pengfei Liu, Yue Zhang, and Zheng Zhang. Ragchecker: A fine-grained framework for diagnosing retrieval-augmented generation. In *Advances in Neural Information Processing Systems (NeurIPS), Track on Datasets and Benchmarks*, 2024.
- [5] Linda Zeng, Rithwik Gupta, Divij Motwani, Yi Zhang, and Diyi Yang. Worse than zero-shot? a fact-checking dataset for evaluating the robustness of RAG against misleading retrievals. *arXiv preprint arXiv:2502.16101*, 2026.
- [6] Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. Building efficient universal classifiers with natural language inference. *arXiv preprint arXiv:2312.17543*, 2024.
- [7] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations. *arXiv preprint arXiv:2102.10073*, 2021.
- [8] Gordon Cormack, Charles Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods, 2009.