No Stupid Questions: Analyzing LLM-generated Queries for Scientific Citation Recommendation

Anonymous ACL submission

Abstract

Each year scientific literature discovery becomes increasingly difficult, making it challenging for researchers to identify contributions relevant to their own work outside of their domain. Scientific recommender systems can make discovery less cumbersome but remain susceptible to exposure biases which could reinforce intra-domain isolation. We use LLMs as a proxy for query generation, assessing their ability to take on diverse roles in assisting with citation recommendation. We compare batches of LLM-generated queries formed as both interrogative complete sentences and sets of keywords and find that interrogative queries yield results which are simultaneously more topically relevant and diverse. We additionally measure the impact of the underlying retrieval system on realizing topicality and diversity in the generated query batches. While our findings indicate that LLMs can produce queries which yield both topical and diverse result sets, we also discuss some design considerations for developing stronger scientific recommender systems that can accommodate the strengths of LLM query generators.

1 Introduction

Each year, researchers are presented with a seemingly insurmountable body of new information, with over half a million new additions to DBLP and 300k new additions to arXiv in 2025 alone. Researchers face a dilemma: spend more time identifying new articles to read, or risk isolation within the enclave of their familiar domain. This problem is exacerbated as scientific participation increases, further imposing "information overload" on research workflows.

Information overload describes a bottleneck in a system where the volume of information available exceeds the means by which to process it (Schneider, 1987). Researchers experience information overload when the set of research articles that are relevant to their interests becomes too large to effectively ingest and synthesize. Academic search engines like Semantic Scholar attempt to ease the process of identifying relevant articles by using semantic search algorithms and personalized recommendation systems. Ricci et al. (2021) motivate recommender systems as remedies to information overload, with the goal of exposing users to novel information that is relevant to their task at hand. Despite the widespread adoption of scientific recommender systems, these tools leave users susceptible to various scholarly biases, such as *exposure bias* (Färber et al., 2023).

Exposure bias can be present in any recommender system which relies on signals derived exclusively from positive-labeled data (Chen et al., 2023). In science, citations are a common example of this. The presence of a citation is considered "proof" of relevance between attribution text and a cited article while uncited articles, by contrast, are presumed to be equally irrelevant. Exposure bias can lead to the emergence of filter bubbles, constraining recommendations exclusively to those which users will likely perceive as positive or familiar (Pariser, 2011; Portenoy et al., 2022; Areeb et al., 2023). Färber et al. (2023) emphasize the importance of explainability in scientific recommender systems to mitigate bias and prevent researchers from being increasingly isolated within independent research silos.

The basis of this paper is to explore citation recommenders for scientific writing through the lens of diversity, limiting the effects of exposure bias. Existing approaches to citation recommendation make the assumption that authors are eminently aware of the best attributable articles as they write. The goal of the retrieval system in this scenario is to spare researchers the trouble of having to recall these articles themselves. As publication numbers have grown to be unwieldy, this expectation has become unrealistic. Motivated by this, we assess

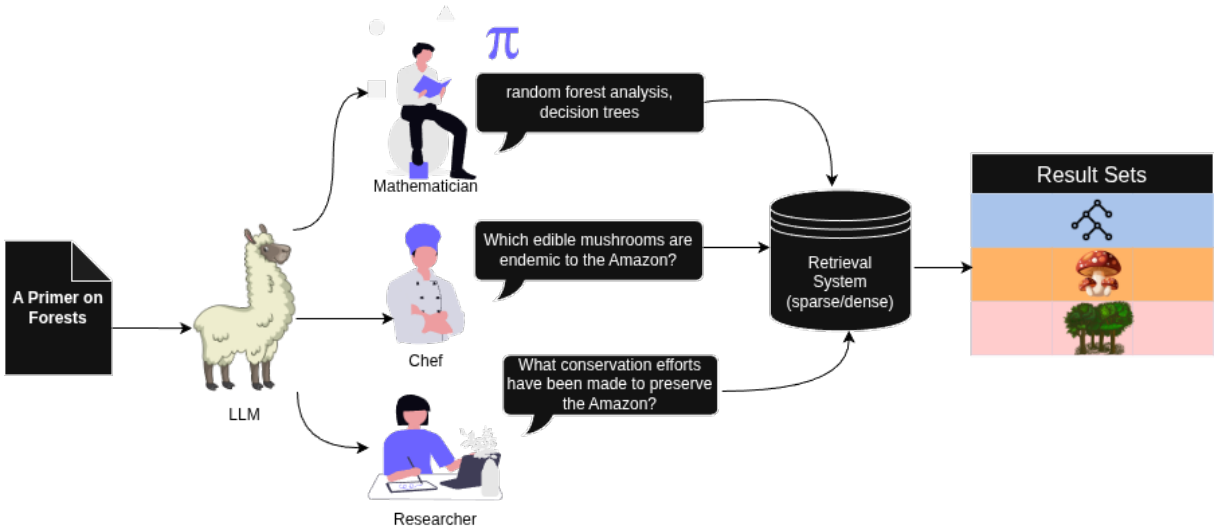

Figure 1: A high-level overview of our retrieval pipeline. An LLM assumes the role of a variety of expert stakeholders, each producing a query to yield diverse result sets.

the potential of LLMs as query generators which emphasize diversity in citation recommendation. Specifically, we use an LLM to generate a diverse set of questions about an article draft with the goal of identifying semantic links which might not be immediately obvious to a researcher. We illustrate a high-level overview of our pipeline in Figure 1.

Consider a case where a researcher is writing an article on problem solving through procedural reasoning. Through the lens of diverse roles, recommenders have the potential to yield suggestions which are equally topical and novel for a researcher while helping her contextualize her own contributions. We motivate this in Table 1, where each role yields a different question with varying expert language and perspective. We contend that LLMs are well-suited for this task because they can retain large amounts of information and perspectives contained in their massive training corpora. Moreover LLMs are reliable natural language interfaces, motivating their utility as components of interpretable recommender systems.

We evaluate this potential application by generating batches of interrogative queries with an off-the-shelf LLM and evaluating their relevance to an article excerpt with respect to two facets: *topicality* and *diversity* (Chen and Xu, 2005; Castells and Jannach, 2023). Topicality is the degree to which results adhere to an underlying topic. In scientific citation recommendation, the topic is a scientific article and its associated metadata. Diversity describes the variety in suggestions put forth by a recommender system. We consider diversity to be equally crucial to the success of a scientific recommender system in mitigating exposure biases.

Through our experiments, we address the following research questions:

RQ1: Are interrogative queries generated by an LLM advantageous to keyword queries with respect to diversity and topicality?

RQ2: How are the diversity and topicality of LLM-generated queries affected by the choice of retrieval system?

RQ3: Which features of LLM-generated queries are crucial for ensuring results that are both novel and topically relevant?

2 Related Work

The task most similar to the experiment detailed in this paper is Local Citation Recommendation (LCR). LCR is a class of citation recommendation techniques which use both the global metadata of an article and a local context excerpt as a query to yield a list of suggestions (He et al., 2010; Jeong et al., 2020; Wang et al., 2025). LCR systems are optimized for precision against a ground truth article which is masked from the local context. While many approaches have relied on classic retrieval pipelines, such as retrieve and rerank (Gu et al., 2022), some recent techniques forego this entirely, opting to use generative language models to reconstruct citations from masked contexts without an oracle database (Çelik and Tekir, 2025).

Title	Prototype-Based Dynamic Steering for Large Language Models
Abstract	Despite impressive breadth, LLMs still rely on explicit reasoning instructions or static, one-fits-all steering methods, leaving a gap for adaptive, instruction-free reasoning amplification. . .
Context	Our method consists of three stages: (1) collecting activation difference vectors representing CoT inducing behavior, (2) discovering a set of "reasoning prototypes" by clustering these difference vectors, and (3) using these prototypes to construct an input-specific steering vector for each input dynamically, during inference. . .

Role	Description	Question
Neural Network Architect	Designs and develops large language models and their components, including their architecture, training methods, and evaluation protocols.	What are the theoretical frameworks that underlie the concept of "activation difference vectors" and how do they relate to the layer-wise extraction of hidden state activation h_l?
Cognitive Scientist	Studies human cognition and the underlying cognitive processes that enable reasoning and problem-solving in humans and artificial intelligence systems.	Can the proposed "reasoning prototypes" be related to cognitive architectures that model human reasoning processes, and how might these prototypes be linked to the concept of "mental models"?
Computational Psychologist	Explores the computational models of cognitive processes, including perception, attention, memory, and decision-making, to better understand human cognition and its applications in AI systems.	How do the results of this study align with the computational models of cognitive processes, particularly in terms of the representation of abstract concepts and the role of attention in shaping reasoning?

Table 1: We use expert prompting (Xu et al., 2023) to generate a diverse set of questions to be used as semantic search queries.

While the experiment detailed in this paper bears some similarity with LCR techniques, we distance our work from LCR as a task because of fundamental limitations to its application. LCR benchmarks model local context as the area both before and after where the citation is needed. While these additional context clues may provide assistance in identifying the missing citation, in practice this is not a recommendation task because authors are aware of the existence of an attributable article when they write sentences which require an attribution. Our work prioritizes the discovery of a diverse set of topically relevant articles which an author can use to better position their work within an article excerpt.

3 Dataset

3.1 Sourcing a Diverse Set of Articles

To fairly assess both topicality *and* diversity across batches of generated queries, we assemble a collection of articles sourced from repositories which cater to different research domains. We use the Semantic Scholar Open Research Corpus (Lo et al., 2020)[1] (S2ORC) as a starting point because it contains full text articles from a variety of sources. Because S2ORC contains millions articles, we restrict the scope of our dataset to a subset of articles sourced from ArXiv, ACL, and PubMedCentral.

[1]We use the 2025-11-04 release of the Semantic Scholar Datasets API.

3.1.1 Retrieval Corpus

We reserve articles which were published before 2025 for our retrieval corpus. In line with prior work on scientific document representation learning (Cohan et al., 2020; Singh et al., 2023), we only consider the title and abstract field for each document in our retrieval corpus. Ultimately, our retrieval corpus consists of approximately 7.6 million titles and abstracts.

3.1.2 Prediction Data

Articles from S2ORC which were published in 2025 were considered for our experiment dataset on the basis of whether or not they contain an eligible citation. A citation is considered eligible if it refers to an article contained in our retrieval corpus. To avoid any one article from being disproportionately represented, we consider a maximum of one eligible citation per article.

For each article, we extract the title, the abstract, and the local context associated with the eligible citation. Several citation recommendation benchmarks define local context as the region *around* a citation marker, bounded by character indexes in each direction (Medić and Snajder, 2020; Gu et al., 2022). Because attending to future information can heavily constrain diversity in text generation (Zimmerman et al., 2025), we instead define local context as the 10 sentences prior to each citation marker.

Name	Type	Description
Query Length (Unigrams)	Surface	Number of words (unigrams) in *query*
Query Length (Tokens)	Surface	Number of Specter2 tokens extracted from *query*
Article Similarity	Content	Cosine similarity between encoded *query* and *(title,abstract)*
Article Unigrams	Content	Unigram overlap between *query* and *(title, abstract)*
Article Bigrams	Content	Bigram overlap between *query* and *(title, abstract)*
Context Unigrams	Content	Unigram overlap between *query* and *local context*
Context Bigrams	Content	Bigram overlap between *query* and *local context*
Query TF-IDF (Mean)	Statistical	Mean tf-idf term vector of *query*
Query TF-IDF (L2-Norm)	Statistical	L2 norm of tf-idf term vector of *query*

Table 2: Predictive features extracted for each query.

4 Queries and Result Sets

Existing citation recommenders for science use a combination of title, abstract, and local context as a query to yield a set of recommendations that emphasize topical relevance rather than diversity. Local citation recommendation systems reinforce this effect by considering the ground truth citation to be the gold standard of recommendation. By instead using a language model to "ask" questions about this content, our aim is to combine the article contents with the model's latent knowledge to reveal recommendations that are competitively topical while also much more novel.

4.1 Expert Roles

In order to produce questions from a variety of diverse perspectives, we leverage the model's latent knowledge by using expert prompting (Xu et al., 2023). Expert prompting is the process of injecting one or more "expert roles" and associated descriptions into a prompt in order to elicit more complex behaviours in generation.

For each article and citation pair, we prompt the model to produce a set of three "expert roles" to gain perspective in query generation. Using the title, abstract, and context, the model is tasked with producing a set of roles which could be valuable consultants in identifying relevant literature. In addition to each role, the model provides a description detailing why such an expert would be relevant to the task.

4.2 Questions and Keywords

Using the produced list of roles, we prompt the model to produce queries in the form of questions and keyword sets, using separate prompts. For each query type, the model should denote which perspective it is selecting first in order to guide the generation of each query. We list examples of the produced roles and question queries in Table 1. For queries composed of keyword sets, we constrain each set to no more than five multi-word terms each.

For each article excerpt, we generate[2] a total of 10 question queries and 10 keyword queries with the intent to evaluate batchwise topicality and diversity. Due to resource constraints, we only produce queries for 500 articles, resulting in a total of 5,000 queries of each type.

4.3 Collecting Result Sets

We issue generated queries to a retrieval system and record the top 10 results. To understand the effects of retrieval system paradigms on topicality and diversity, we issue each query to a sparse index and a dense index for comparison.

4.3.1 Dense Retrieval

To build our dense index, we first encode each document in the retrieval corpus using Specter2 (Singh et al., 2023). The index itself is constructed by loading the pre-computed embeddings into FAISS (Douze et al., 2024). At retrieval time, we encode the generated queries using the Specter2 `adhoc_query` adapter and issue it to FAISS. From the position of the encoded query in the embedding space, we record the 10 closest documents by Euclidian distance for our analysis.

4.3.2 Sparse Retrieval

Our sparse index is built with Elasticsearch [3] using the Lucene implementation of the BM25 ranking function (Robertson and Zaragoza, 2009). We insert the titles and abstracts of the retrieval corpus documents into an Elasticsearch *text* field. Elasticsearch uses an analyzer to parse each document and build an inverted index, recording the frequency of terms as they appear across the entire corpus. At retrieval time, the generated queries are parsed with

[2] We used Llama-3.2-3B-Instruct for query generation.
[3] https://www.elastic.co/elasticsearch

the same analyzer and the 10 documents which are ranked highest by the BM25 algorithm are recorded for our analysis.

5 Experimental Setup

Our research questions explore not only the differences between types of generated queries but also the effects of different retrieval paradigms on topicality and diversity. We first identified a set of predictor features to capture variance in query structure and overlap with the source articles used to generate them.

5.1 Predictor Features

Predictor features are selected on the basis of their interpretability and controllability during query generation We organize them into three categories: surface features, content features, and statistical features.

Surface features Surface features describe attributes of the queries in isolation, such as unigram length and token length. Generally, concise keyword terms tend to be much shorter than questions formed as complete sentences.

Unigram length is collected using the NLTK word tokenizer, excluding punctuation tokens. Token length is the number of tokens produced when tokenizing the queries using the Specter2 tokenizer from HuggingFace.

Content features Content features describe the relationship between the queries and the articles that were used to generate them. These relationships can be semantic or lexical.

We quantify semantic overlap by measuring the cosine similarity between the encoded queries and their respective articles. High semantic overlap indicates loose topical similarities without necessarily using similar words.

Additionally, we record lexical measures of overlap by counting the unigram and bigram matches between the query and the title, abstract, and context used to produce it.

Statistical Features Statistical features describe queries by their statistical relationship with the retrieval corpus. Specifically, we measure the tf-idf (Sparck Jones, 1972) of each token in the query and produce two aggregate scores: mean and L2 norm.

Queries and articles alone are not enough to estimate the informativeness of a term. Consider the term "citation-informed transformer" against the semi-synonymous proper noun "SPECTER". The former is composed of tokens `citation`, `informed`, `transformer` which are much more likely to occur frequently in the retrieval corpus and thus have a lower aggregate tf-idf score. We record tf-idf as a proxy for query utility against the retrieval corpus.

5.2 Outcome Measures

We assess topicality and diversity in the retrieved result sets across a set of outcome measures.

5.2.1 Topicality

Informally, topicality is the quantification of the topical relationship between an anchor document and a set of result documents. We consider the topicality of a query to be the utility of a query in connecting the anchor document from which it was generated with topical results via a retrieval system. Concretely, we model topicality as the relationship between an article D, a query $q_i^D \in Q^D$, and a result set R_i^D. The LLM, in generating q_i^D, can use it as a proxy for D by articulating an information need relevant to D. We evaluate the utility of this proxy by measuring the topical relationship between the documents in R_i^D and D as an anchor. We establish two dimensions of topical relevance with D: global and local.

Global Relevance Global relevance describes the relevance between a result document $r \in R_i^D$ and the global metadata of D, specifically the title and abstract. Titles and abstracts contain a bird's-eye view of an article and high similarity between the global metadata of two articles will often entail a topical overlap. `Article-Result Similarity` is the cosine similarity between the Specter2 embeddings of D and each document $r \in R_i^D$. Because each result set contains 10 documents, we aggregate scores with both mean (Eq. 1) and rank-weighting (Eq. 2) for analysis.

$$\text{Article Sim}_{\text{mean}}(d, R) = \frac{1}{|R|} \sum_{r \in R} \cos(d, r_i) \quad (1)$$

$$\text{Article Sim}_{\text{weighted}}(d, R, p) = \sum_{i=1}^{|R|} p^i \cdot \cos(d, r_i) \quad (2)$$

Local Relevance Local relevance is the relevance between a result document $r \in R_i^D$ and the local context c, the sentences extracted from the full text of D, immediately preceding a citation. Local contexts are more reliable anchors of relevance, often containing specific information about motivation or methodology that are too verbose for an abstract.

To evaluate topical relevance between a context anchor and a result document, we consider the case where the context itself were used as a query instead of an LLM-generated query. Inspired by (Gu et al., 2022), we finetune a SciBERT (Beltagy et al., 2019) crossencoder Θ for relevance scoring using 20,000 articles from Semantic Scholar previously excluded from our experimental dataset. Each context contains 5 sentences leading up to, but not including, a sentence containing a citation. Positive datapoints paired contexts with the title and abstract of the article that would be cited in the subsequent sentence. We used BM25 to mine hard negative examples, yielding 80,000 datapoints in total. We trained the model for 3 epochs, ultimately achieving an F1 score of 0.71.

We use the crossencoder Θ to quantify the topical relationship between context c and result documents $r \in R_i^D$. Crossencoders provide a stronger guarantee of topical relevance because the model can cross-attend between the contents of c and the contents of r and produce one scalar score. We report these scores as `Context Crossencoder`, again aggregating the scores with both mean (Eq. 3) and rank-weighting (Eq. 4).

$$\text{Crossencoder}_{\text{mean}}(c, R) = \frac{1}{|R|} \sum_{r \in R} \Theta(c, r_i) \quad (3)$$

$$\text{Crossencoder}_{\text{weighted}}(c, R, p) = \sum_{i=1}^{|R|} p^i \cdot \Theta(c, r_i) \quad (4)$$

5.2.2 Diversity

Diversity in retrieval systems is often described as the variety of non-redundant information in a result set (Castells and Jannach, 2023). For example, intra-list dissimilarity (Ziegler et al., 2005) uses a diversity function z to measure dissimilarity between elements of one list. We extend this definition to evaluate batches of queries generated from the same source article by averaging pairwise dissimilarity of their result sets (Eq. 5).

$$\text{Diversity}(R) = \frac{2}{|R|(|R|-1)} \sum_{i}^{|R|} \sum_{j>i}^{|R|} z(r_i, r_j) \quad (5)$$

Intuitively, when a set of queries Q address a distinct set of information needs, there should be minimal information overlap in their result sets R. We record pairwise set dissimilarity with a selection of inverted overlap measures z of varying strictness.

Inverted Set Overlap The most strict measures of overlap use the cardinality of the intersection of two lists. Because our result sets are all equally sized, we record the inverted `Overlap Coefficient` between each result set (Eq. 6).

$$z(r_i, r_j) = 1 - \frac{|r_i \cap r_j|}{\min(|r_i|, |r_j|)} \quad (6)$$

In retrieval experiments, result sets are also ordered where documents higher in rank-ordering are presumed to be of greater relevance. Measures such as Rank-Biased Overlap (RBO) (Webber et al., 2010) account for discrepancies in ordering between lists, favoring overlap at higher ranks to overlap at lower ranks. We measure the inversion of `RBO` between result sets (Eq. 6), favoring dissimilarity at higher ranks where documents are presumably more relevant to the query.

$$z(r_i, r_j) = 1 - \left[(1-p) \sum_{k=1}^{|r|} p^{k-1} \frac{|r_i[:k] \cap r_j[:k]|}{k} \right] \quad (7)$$

Many implementations of RBO include a parameter p which, when 0, produces uniform weights and, when 1, considers only the highest ranked document. We use a standard $p = 0.5$ for our experimentation.

Inverted Similarity To assess semantic diversity between result sets we extend cosine similarity to compare each constituent document $a \in r_i$ with each $b \in r_j$, measuring the average pairwise `Result Set Similarity` (Eq. 8)

$$z(r_i, r_j) = 1 - \frac{1}{|r_i||r_j|} \sum_{a \in r_i} \sum_{b \in r_j} \cos(a, b) \quad (8)$$

We additionally compare the set of articles c_i cited by the highest ranked result in r_i with the set

Feature	DQ vs DK		DQ vs SQ		DK vs SK		SQ vs SK	
	$\tilde{x}_{DQ}$	$\tilde{x}_{DK}$	$\tilde{x}_{DQ}$	$\tilde{x}_{SQ}$	$\tilde{x}_{DK}$	$\tilde{x}_{SK}$	$\tilde{x}_{SQ}$	$\tilde{x}_{SK}$
Topicality								
`Article-Result Sim.` (mean)	0.8783	**0.8838***	0.8783	**0.8903***	0.8838	**0.8937***	0.8903	0.8937
`Article-Result Sim.` (rank-weighted)	0.8737	**0.8809***	0.8737	**0.8850***	0.8809	**0.8888***	0.8850	0.8888
`Context Crossencoder` (mean)	**0.7320***	0.7159	**0.7320***	0.4312	**0.7159***	0.4550	0.4312	**0.4550***
`Context Crossencoder` (rank-weighted)	**0.7832***	0.7623	**0.7832***	0.4588	**0.7623***	0.4946	0.4588	**0.4946***
Diversity (Inverted Overlap)								
`Overlap Coefficient`	**0.9500***	0.9111	0.9500	**0.9666***	0.9111	**0.9277***	**0.9666***	0.9277
`RBO`	**0.9873***	0.9623	0.9873	**0.9959***	0.9623	**0.9746***	**0.9959***	0.9746
`Result Set Similarities`	**0.1415***	0.1369	**0.1415***	0.1325	**0.1369***	0.1272	**0.1325***	0.1272
`Result Citation Similarities`	**0.1982***	0.1964	0.1982	0.1986	**0.1964***	0.1938	**0.1986***	0.1938

Table 3: Median result values across outcome measures. Asterisks denote in-group statistical significance.

of articles c_j cited by the highest ranked result in r_j to have a better understanding of the bibliometric topology of each result set. The citations of the highest ranked document in each result set serve as a proxy for the bibliometric identity for the rest of the result set (Eq. 9).

$$z(c_i, c_j) = 1 - \frac{1}{|c_i||c_j|} \sum_{a \in c_i} \sum_{b \in c_j} \cos(a, b) \quad (9)$$

By recording the inverse of the `Result Citation Similarity`, we can observe result sets which diverge with respect to bibliometric coupling.

6 Analysis and Discussion

We evaluate four configurations of query structure and retrieval paradigm: Dense Question (DQ), Dense Keyword (DK), Sparse Question (SQ), and Sparse Keyword (SK). The comparison of DQ×DK isolate the effects of query structure in LLM-generated queries while DQ×SQ and DK×SK examine the effects of the retriever on our topicality and diversity measures.

We report the difference in results of the outcome measures within these groups in Table 3. Topicality features are measured independently per-query while diversity features are computed with batch-wise averaging. To accommodate this, we apply the Mann-Whitney U test for topicality features and we use the Wilcoxian signed-rank test over the batchwise medians for the diversity features.

6.1 RQ1: Query Structure

Overall, we found that queries formed as questions carried greater retrieval utility with respect to both topicality and diversity. While the DK group had slightly higher `Article-Result Sim` biencoder scores, the DK group performed much worse than DQ on `Context Crossencoder`. While this could indicate that LLMs can indeed articulate the complex information needs of an article excerpt more easily when writing questions, it could also indicate that the cosine similarity between two precomputed embeddings is not a reliable measure of semantic similarity. The difference in these scores could also be a testament to the information density between titles, abstracts and contexts. While titles and abstracts are usually constrained by a maximum number of words, contexts have much more space to establish a narrative and may contain less clusters of semantically rich text.

We observed an entirely different pattern between SQ and SK, where keyword queries generally yielded results which were more topically relevant to the context but question queries were associated with more diversity in recommendation. We initially suspected that this behaviour could be attributed to the longer length of question queries, on average, which would dilute their utility. However, upon doing a post-hoc correlative analysis between query length and the outcome measures, we found no evidence to suggest this.

6.2 RQ2: Retrieval System

While sparse retrievers are optimized for probabilistic exact term matching, our results suggest that they may be limited in their ability to provide suggestions that are both topical and diverse simultaneously. While exact term matching is valuable when the user of a retrieval system is knowledgeable enough of their domain to formulate precise queries, it is a fundamental limitation in scenarios where a researcher may want to find articles beyond

Predictor	Topicality Features				Diversity Features			
	Article-Result Similarity (mean)	Article-Result Similarity (weighted)	Context Crossencoder (mean)	Context Crossencoder (weighted)	Overlap Coefficient	RBO	Result Set Similarity	Result Citation Similarity
Query Length (Unigrams)	0.0005	0.0003	-0.0053	-0.0062	-0.0040	-0.0037	-0.0013	-0.0003
Query Length (Tokens)	-0.0002	-0.0002	0.0001	0.0006	0.0016	0.0011	0.0003	0.0002
Article Similarity	**0.3619**	**0.3751**	**-0.1744**	**-0.2311**	**-0.2423**	**-0.2166**	**-0.1863**	**-0.1458**
Article Unigrams	0.0343	0.0328	0.0848	0.0788	-0.0272	-0.0246	-0.0026	-0.0062
Article Bigrams	0.0003	0.0011	-0.0625	-0.0389	**-0.1159**	**-0.1246**	-0.0208	-0.0263
Context Unigrams	0.0233	0.0209	**-0.4686**	**-0.4306**	-0.0628	-0.0057	-0.0494	-0.0304
Context Bigrams	-0.0217	-0.0194	**0.1902**	**0.1641**	-0.0411	-0.0680	0.0423	0.0480
Query TF-IDF (mean)	0.0001	-0.0001	-0.0023	-0.0002	-0.0114	-0.0116	-0.0015	0.0003
Query TF-IDF (L2-Norm)	0.0006	0.0007	-0.0001	-0.0003	-0.0053	-0.0035	-0.0006	-0.0006

Table 4: Regression coefficients for topicality and diversity features.

their expertise without knowing the appropriate scientific jargon. The low *Context Crossencoder* scores may reflect the LLM's lack of exposure to the retrieval corpus. Without access to an oracle of domain-specific knowledge or the ability to peruse the corpus itself, an LLM may struggle forming lexically informative queries for a sparse scientific retriever.

While the sparse retriever groups yielded poor topicality scores, between them there was a notable difference in diversity scores. The sparse retriever yielded significantly more diverse results, indicating less overlap between the results of the question queries.

6.3 RQ3: Feature and Outcome Correlation

We conducted a regression analysis between predictor features and outcome measures in an effort to identify which qualities of a query are most tightly coupled with our desired effects. While some predictors yielded intuitive relationships with certain outcome measures, others were surprising. We report these results in Table 4.

Perhaps least surprisingly, semantic similarity between the article and the query is moderately correlated with semantic similarity between the article and the result sets. Notably, similarity between the query and the article is negatively correlated with all diversity measures. One possible explanation for this could be that some article contents more tightly constrain query generation, causing the queries to yield result sets with both high overlap and high semantic similarity with the article. This is further reinforced by the negative correlation between article bigram overlap and both inverted overlap coefficient and RBO. Highly specific bigrams may pose too strong of a constraint on query generation when using article excerpts as a seed.

We observe this same effect with the `Context Crossencoder` scores. While crossencoder scores are negatively correlated with query-context unigram overlap, the bigram overlap tells a different story. Although bigrams are effectively an extension of unigrams, they can capture additional positional information which unigrams do not. With this positional data, bigrams can effectively encode small amounts of semantic information which contain much richer information than their constituent unigrams in isolation. Consider the bigrams in this article itself, such as "citation recommendation" or "generated queries", which encode topical meaning that would be otherwise lost if the terms were considered separately.

7 Conclusion

We design an experiment to assess the utility of an LLM in query generation for scientific recommender systems. Overall, we find that LLMs can produce queries which yield more topically relevant and diverse recommendations while generating interrogative queries as opposed to sets of keyword search terms. We also identify limitations of sparse retrievers in the realization of generated query batches with respect to both topicality and diversity simultaneously. Through regression analysis, we find relationships between predictors and our proposed outcome measures which can offer some design considerations in developing future retrieval systems for scientific citation recommendation.

Limitations

We have identified two limitations of our work, each with respect to our assembled dataset.

The first limitation is that there may be instances in which data is incomplete due to Semantic Scholar having an imperfect index. We've found multiple instances where the S2ORC citation annotations contained citations which we missing when fetching the complete document from Semantic Scholar by its corpus id. This could introduce noise to our evaluation which already only spans a small 500 documents.

The second limitation is regarding the applicability of our methods to article sources beyond ACL, ArXiv, and PubMedCentral (the three which we used). While we deliberately chose these sources for a diverse collection of documents, they are somewhat insular and the sparsity in their embeddings could simply be too much to observe any effects between domains.

References

Qazi Mohammad Areeb, Mohammad Nadeem, Shahab Saquib Sohail, Raza Imam, Faiyaz Doctor, Yassine Himeur, Amir Hussain, and Abbes Amira. 2023. Filter bubbles in recommender systems: Fact or fallacy—A systematic review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(6):e1512.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3615–3620.

Pablo Castells and Dietmar Jannach. 2023. Recommender systems: A primer. *arXiv preprint arXiv:2302.02579*.

Ege Yiğit Çelik and Selma Tekir. 2025. Citebart: learning to generate citations for local citation recommendation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1703–1719.

Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems*, 41(3):1–39.

Zhiwei Chen and Yunjie Xu. 2005. User-oriented relevance judgment: A conceptual model. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 101b–101b. IEEE.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 2270–2282.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Michael Färber, Melissa Coutinho, and Shuzhou Yuan. 2023. Biases in scholarly recommender systems: impact, prevalence, and mitigation. *Scientometrics*, 128(5):2703–2736.

Nianlong Gu, Yingqiang Gao, and Richard HR Hahnloser. 2022. Local citation recommendation with hierarchical-attention text encoder and scibert-based reranking. pages 274–288. Springer.

Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, and Lee Giles. 2010. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 421–430.

Chanwoo Jeong, Sion Jang, Eunjeong Park, and Sungchul Choi. 2020. A context-aware citation recommendation model with bert and graph convolutional networks. *Scientometrics*, 124(3):1907–1922.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S Weld. 2020. S2orc: The semantic scholar open research corpus. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4969–4983.

Zoran Medić and Jan Snajder. 2020. Improved local citation recommendation based on context enhanced with global information. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 97–103, Online. Association for Computational Linguistics.

Eli Pariser. 2011. *The filter bubble: How the new personalized web is changing what we read and how we think*. Penguin.

Jason Portenoy, Marissa Radensky, Jevin D West, Eric Horvitz, Daniel S Weld, and Tom Hope. 2022. Bursting scientific filter bubbles: Boosting innovation via novel author discovery. pages 1–13.

Francesco Ricci, Lior Rokach, and Bracha Shapira. 2021. Recommender systems: Techniques, applications, and challenges. *Recommender systems handbook*, pages 1–35.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc.

Susan C Schneider. 1987. Information overload: Causes and consequences. *Human Systems Management*, 7(2):143–153.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. Scirepeval: A multi-format benchmark for scientific document representations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5548–5566.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Qianqian Wang, Hao Li, Mingjie Ma, and Zhenhua Li. 2025. A context-aware enhanced local citation recommendation model integrating scibert and self-adaptive attention. *Scientometrics*, 130(8):4495–4517.

William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38.

Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. 2023. Expertprompting: Instructing large language models to be distinguished experts. *arXiv preprint arXiv:2305.14688*.

Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32.

Brian Zimmerman, Gaurav Sahu, and Olga Vechtomova. 2025. Future sight: Fine-tuning language models for dynamic story generation. In *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*, pages 233–248. Springer.