Internship Report

How to use generative models to improve focused crawling ?

National Center for Scientific Research "Demokritos"

Apolline Guérineau

Supervisor : George Giannakopoulos

2023-2024

Abstract

Focused crawling aims to enhance the efficiency of web crawlers by directing them to gather content specifically related to a particular topic, rather than exhaustively indexing the entire web. In this context, focused crawling can significantly support researchers and students looking to find relevant papers within their field of study. The goal of this paper is to evaluate several techniques that could improve the ability of a focused crawler to retrieve relevant research papers, namely through query expansion using Large Language Models (LLMs) and by employing the HYDE method for similarity calculation.

In order to test these techniques, we designed three distinct use cases, simulating the information-seeking behavior of a researcher using focused crawling systems. In summary, our findings suggest that query expansion via an LLM has the potential to uncover relevant papers that would otherwise be missed, though further refinements in prompt design are necessary to maximize its effectiveness. The classifiers, while useful in some contexts, do not consistently improve results and, in some cases, may perform worse than the default search engines. The HYDE method shows promise and could play a key role in improving focused crawling techniques.

Code is available at https://github.com/apollineguerineau/demokritos_internship.git

Table of contents

Introduction **1**

1 Proposed Focused Crawling Configurations **3**

1.1 Frameworks overview 3

1.2 Execution Workflow 5

2 Evaluation Procedure for Focused Crawling Configurations **6**

2.1 Use Cases to evaluate the focused crawling configurations 6

2.2 Evaluation dataset 7

2.3 Analysis of Focused Crawling Configurations 8

3 Results **10**

3.1 Use case 1 : RAG AND «code generation» 10

3.2 Use case 2 : "metal-organic frameworks" AND "material design" AND "properties" . . . 14

3.3 Use case 3 : "Machine Learning" AND (diffusion OR diffusivity) AND (MOFs OR ZIFs OR "metal-organic frameworks" OR COFs OR "covalent-organic frameworks) 18

3.4 Qualitative Evaluation of Expanded Queries and HYDE 18

Conclusion **19**

Bibliography **20**

Appendix **21**

Introduction

Unlike traditional web crawlers that aim to explore the web exhaustively, focused crawling is designed to gather web pages specifically relevant to a particular topic. Introduced in 1999 by the paper [1], this targeted approach saves resources while enhancing the efficiency of information retrieval. Focused crawlers are particularly valuable for researchers, as they address a critical need within the scientific and academic community: the effective indexing and discovery of research papers. Although online databases and archives exist to catalog these papers, their search mechanisms are not always optimal, often leading to retrieval gaps or difficulties in accessing niche or emerging topics.

There are two main approaches to focused crawling. The first, search-based focused crawling, involves submitting specific keywords or phrases to search engines to retrieve a list of pages potentially relevant to a given domain. The second, crawling-based focused crawling, extracts links from already discovered pages and follows them to identify new pages within the same domain. A focused crawling system typically consists of two main components. The first is a classifier, responsible for evaluating the relevance of documents by assigning them a score. The system's effectiveness strongly depends on the quality of this classifier, as a weak or inaccurate classifier reduces the chances of discovering new relevant pages. However, designing a robust classifier is a complex task, requiring access to a large, labeled dataset that indicates whether pages are relevant or not. Such datasets, which are often expensive to produce, are essential for training the model effectively. More over these trained classifier are domain-dependent and can not be adapt to other subjects. To address this challenge, some studies propose using similarity-based classifiers, which compare newly discovered pages with "seed pages" (provided initially by the user) or directly with the initial query [2]. The second component is a distiller, which plays a role in prioritizing exploration. This component identifies hubs, or pages that centralize numerous links to potentially useful resources. These hubs help guide the search more effectively and efficiently. Recent works have sought to enrich these systems by integrating advanced techniques such as active learning [3], which optimizes data annotation by focusing on uncertain examples, and reinforcement learning [3, 4], which improves exploration by assigning rewards based on the relevance of visited pages. Another critical dimension is query expansion, particularly useful for search-based approaches. Keywords related to a domain are often too general or lack context, which can lead to irrelevant results. Query expansion aims to refine these keywords to ensure more precise and targeted results [4, 2, 5]. For example, traditional methods like Pseudo-Relevance Feedback (PRF) generate expanded queries by extracting keywords from documents retrieved in previous iterations.

This work focuses on the dimension of query expansion, a topic widely studied in information retrieval. Information retrieval involves identifying relevant documents from a database based on a user query. Traditional techniques, such as BM25 (Best Match 25), rely on statistical matching, while more recent approaches leverage embeddings to represent queries and documents as vectors, facilitating similarity calculations. Despite their utility, these methods have limitations. Similarity calculations rely solely on textual or semantic proximity, which can be insufficient when the user query is too short to provide adequate context, is ambiguous, or refers to specialized vocabulary. Query expansion methods address these challenges by enriching the initial query to improve similarity calculations. Traditional approaches like PRF retrieve an initial set of documents deemed relevant (typically the top k results from a search) and extract key terms to expand the original query. The goal is to integrate frequently co-occurring terms from the relevant results to improve recall. However, these traditional methods assume that the initially retrieved documents are relevant, which is not always the case, particularly for short, ambiguous, or poorly formulated queries. If the initial results are of low quality, PRF can introduce irrelevant terms, further degrading search results. To overcome these limitations, recent studies, such as [6] and [7], explore the use of Large Language Models (LLMs), like GPT, for query expansion. These models possess advanced generative capabilities that allow for more contextualized and nuanced

query enrichment. By using prompts to guide the model, LLMs can generate expanded queries based on the initial query, addressing shortcomings of traditional approaches.
This work explores two types of query expansion: the direct expansion of the initial query to improve the results of search-based focused crawlers, and the generation of a hypothetical relevant document (Hyde) to enhance similarity calculations for the classifier. The Hyde method, introduced in [8], produces a document that ideally represents the expected content in response to a query, facilitating better matches with available documents.

1 Proposed Focused Crawling Configurations

1.1 Frameworks overview

The focused crawling configurations implemented in this work are designed to operate with ArXiv and ChemRxiv. Both platforms are prominent online repositories for research papers, serving as essential resources for the academic and scientific community. ArXiv focuses on a wide range of domains, including physics, computer science, and mathematics, while ChemRxiv specializes in chemistry. These platforms enable researchers to share their findings and access the latest developments in their areas of interest. Both repositories provide APIs (Application Programming Interfaces) that allow users to programmatically query their databases. By leveraging their respective APIs, the focused crawling system retrieves key metadata such as the URL, title, and abstract, which are then analyzed to identify relevant documents.

We consider a user perspective where the user can either be experienced (they are familiar with the domain of their work) or a beginner (seeking papers to gain knowledge about the domain). We assume that such a user will only browse through the **top 50 papers** retrieved. The goal of this study is to develop a system that improves recall (by increasing the number of relevant documents) and precision (by decreasing the number of irrelevant documents) within these top 50 papers.

To achieve this, we employ query expansion systems and relevance scoring systems.

Two query expansion systems were implemented:

- Query expansion based on the initial query: The following template is provided to the LLM to request a query expansion:
  "'Your task is to expand the following search query. Generate an improved query by adding one or more related terms, synonyms, or key variations that reflect a broader or more specific focus based on the original query. Your goal is to optimize the query for retrieving more relevant search results.
  This is the initial query:
  Please generate a new expanded boolean query. The response must be in JSON format, with the key "expanded_query" and for value the suggested expansion. Just output the JSON."'
  Configurations using this system are marked with the notation *seed-query*.

- Query expansion based on the title of the most relevant paper found so far: The following template is provided to the LLM in this case:
  "'Your task is to expand the following search query based on additional information from a relevant page title. Start by analyzing the initial query and then consider how the title provided can enhance the context. Generate an improved query by adding one or more related terms, synonyms, or key variations that reflect a broader or more specific focus based on both the original query and the page title. Your goal is to optimize the query for retrieving more relevant search results.
  This is the initial query:
  And this is the relevant page title:
  Please generate a new expanded boolean query that integrates these insights. The response must be in JSON format, with the key "expanded_query" and for value the suggested expansion. Just output the JSON."'
  Configurations using this system are marked with the notation *best-paper*.

The LLM used to generate new queries is Llama3.2, released in September 2024 by Meta.

Two relevance scoring systems were implemented:

- Cosine similarity between the initial query and the papers, using the union of their title and description.
  To this end we use the embedding model e5 that performs well on most benchmarks. For each comparison between the initial query and a paper, first the text is encoded by the embedding model, then the cosine similarity is computed. This is explained by the figure 1.1 :

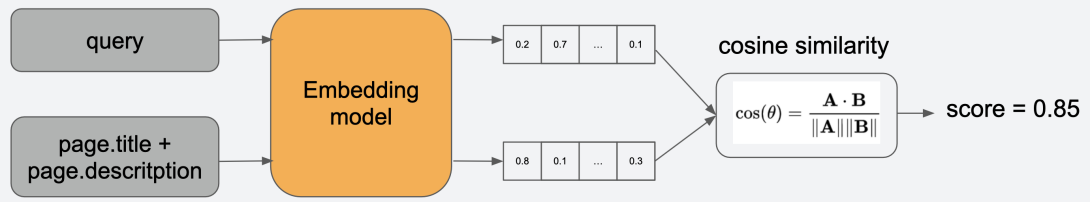


Figure 1.1: Cosine similarity between initial query and a paper

  Configurations using this system are marked with the notation *sim-cos*.

- Cosine similarity between a Hyde and the papers : This involves generating a Hyde paper. To this end, we ask an LLM to generate a 'fake' (hypothetical) paper consisting of a title and an abstract. The following prompt was used:
  "'Your task is to generate a research paper title and abstract based on a search query I will provide. The title should be precise, engaging, and clearly reflect the main topic. The abstract should provide a clear overview of the objectives, methodology, and implications of the research. Please output your response in json format with keys 'title' and 'abstract'.
  This is the search query : "'
  The LLM used here is again Llama3.2.
  Similarity is calculated in the same way as above. Configurations using this system are marked with the notation *hyde-sim-cos*.

Some configurations also utilize an initial query description. In this case, the user is asked to provide a brief description explaining their query and the principles of the domain. This description can then be used for two different purposes:

- Improvement of Hyde generation: The description is included in the prompt sent to the LLM for generating the Hyde. This is expected to generate higher-quality Hydes that better align with the user's expectations. As a result, the scoring system could perform better.
- Improvement of query expansion: The description is included in the prompt sent to the LLM for generating an expanded query. This can potentially produce queries more closely aligned with the user's needs.

Configurations using an initial query description are marked with the notation *description*.

In total, we have 6 configurations:

- `baseline`: No query expansion or relevance scoring systems. Simply returns the top 50 documents as provided by the API. This represents the documents the user would examine without a focused crawling system.
- *seed-query_sim-cos*

- *seed-query_hyde-sim-cos*
- *seed-query_hyde-sim-cos_description*
- *best-paper_hyde-sim-cos*
- *best-paper_hyde-sim-cos_description*

1.2 Execution Workflow

The execution process begins with the initial query, where all documents returned by the API are retrieved and scored using the designated scoring system. Following this, a new query is generated through the query expansion system, and the documents returned for this query are also retrieved and scored. This iterative process continues, until the configuration has completed the retrieval and scoring of documents for a total of **10 queries**. This workflow ensures a progressive exploration of the document space while incorporating the effects of query expansion and scoring mechanisms at each step.

2 Evaluation Procedure for Focused Crawling Configurations

2.1 Use Cases to evaluate the focused crawling configurations

Three use cases were identified:

Use Case 1: ***"RAG AND Code Generation"*** : The query is appropriate but does not return all relevant documents.

- API: ArXiv
- The original dataset—representing results without focused crawling—contains only 17 relevant documents out of 24 total. The goal of focused crawling here is to uncover additional relevant documents. The main question evaluated is: *"Does the system provide the user with papers they would not have found on their own?"*

Use Case 2: ***"metal-organic frameworks" AND "material design" AND "properties"*** : The user is a novice researching the topic, which relates to the term "inverse design," though the user is unaware of this term.

- API: ArXiv
- The original dataset contains only 9 papers, of which 7 are relevant. The goal of focused crawling is to provide the user with more relevant documents. Again, the main question evaluated is: *"Does the system provide the user with papers they would not have found on their own?"*

Use Case 3: ***"Machine Learning" AND (diffusion OR diffusivity) AND (MOFs OR ZIFs OR "metal-organic frameworks" OR COFs OR "covalent-organic frameworks)*** : The user is an expert in their domain, and their initial query is "perfect," as it already retrieves all relevant documents.

- API: ChemRxiv
- The original dataset contains 6083 papers, of which only 5 are relevant. We make the reasonable assumption that the original dataset already includes all relevant documents, given the rarity of papers on this highly specific topic. The focused crawling system will not uncover new relevant documents. Instead, the evaluation here focuses solely on the relevance scoring system, with the main question being: *"Does the system save the user time?"* Specifically, the system should aim to include the maximum number of relevant documents (ideally all 5) while minimizing the number of irrelevant documents returned.

  Regarding the annotation of the original dataset: the last relevant document annotated is located at line 135. The expert annotated up to line 480, finding only irrelevant documents thereafter. Given the high specificity of the query, relevant documents are exceedingly rare. It was assumed that the remaining documents (from line 480 to 6083) are irrelevant.

2.2 Evaluation dataset

The construction of the evaluation dataset involves two stages of paper labeling.

First Labeling Step: Original Dataset

The user selects an initial query, which is the query they would typically submit to a search engine or an archive database such as ChemRxiv or ArXiv. Using this query, we obtain an initial dataset of papers, which the user is asked to label. Each row in the dataset corresponds to a paper, represented by its URL, title, and description. Based on title and description, the user assigns a label as follows:

- '1' if the paper is relevant to their query.
- '0' if it is not relevant.

This dataset is referred to as the **'original dataset'** for easy reference in later stages.

Table 2.1 is an example of the first three rows of an original dataset for the use case 2 :

URL	Title	Description	Relevance
http://arxiv.org/abs/...	Synthesis and Tailored Properties...	Porous polymeric covalent organic frameworks (COFs) have been under intense synthetic investigation with over 100 unique structural motifs known...	0
http://arxiv.org/abs/...	Optimal pre-train/fine-tune strategies...	Overcoming the challenge of limited data availability within materials science is crucial for the broad-based applicability of machine learning within...	1
http://arxiv.org/abs/...	Structural transformations in porous glasses under mechanical loading.	The evolution of porous structure and mechanical properties of binary glasses under tensile loading were examined using molecular dynamics simulations...	0

Table 2.1: Template of an Original Dataset

Second Labeling Step: Labelled Dataset

The **labelled dataset** is used to evaluate how recall and precision evolve across the **10 queries** for each configuration. For a configuration$_i$, we create a dataset$_i$ by following the procedure below:

- **Query** q_0:
  - Retrieve results for q_0 only.
  - Select the **top 50 documents** from the results, sorted in descending order of score.
- **Query** q_1:
  - Merge the results of q_0 with those of q_1.

- Sort the combined documents ($q_0 \cup q_1$) in descending order of score.
- Select the **top 50 documents** from the ranking.

- **Query** q_2:
  - Merge the results of q_0, q_1, and q_2.
  - Sort the combined documents ($q_0 \cup q_1 \cup q_2$) in descending order of score.
  - Select the **top 50 documents** from the ranking.

- ...

- **Query** q_{10}:
  - Merge the results of q_0, q_1, ..., q_9, and q_{10}.
  - Sort the combined documents ($q_0 \cup q_1 \cup ... \cup q_{10}$) in descending order of score.
  - Select the **top 50 documents** from the ranking.

The set of selected documents constitutes dataset$_i$. Each iteration thus includes not only the current query's results but also those from previous queries. The evaluation aims to measure whether better results are achieved progressively with each query.

Finally, the datasets from all 6 configurations are combined to create the labelled dataset:

$$\textbf{labelled dataset} = dataset_0 \cup dataset_1 \cup ... \cup dataset_5$$

Duplicates and papers already annotated in the original dataset are removed to prevent multiple annotations of the same document, in order to save time. The user is then asked to label each row in the same way as for the original dataset:

- '1' if the paper is relevant to their initial query.
- '0' if it is not relevant.

The original dataset, as described in Section 2.2.1, serves as the starting point for evaluation. This dataset contains a total of N_{total} documents, of which N_{relevant} are relevant to the user's initial query. The goal of the focused crawler is either to identify additional relevant documents that were not part of the original dataset or to keep all relevant ones, both while minimizing the retrieval of irrelevant ones. The original dataset can be then used to evaluate recall and precision among the papers that a user could have found by himself with his initial query.
The labelled dataset constructed as described in Section 2.2.2, consolidates the top 50 documents with the highest scores from the cumulative datasets for each query of all configurations.
We finally construct the **final dataset** as the union of the original dataset and the labelled dataset. We assume that all relevant papers for the considered use case are present in the final dataset. All other documents are considered non-relevant, providing a reference for precision and recall analyses.

2.3 Analysis of Focused Crawling Configurations

This section details the analyses conducted to evaluate the performance and behavior of the proposed focused crawling configurations.

The results analyzed in this study evaluate the performance of different focused crawling configurations based on their ability to retrieve, rank, and retain relevant documents throughout a series of 10 queries. The evaluation examines three key aspects: the total number of relevant documents retrieved, the overlap and diversity of relevant documents found by the different configurations, and the ranking effectiveness of the scoring mechanisms. Additionally, query-by-query performance was assessed to monitor the progressive improvement of the focused crawler, with a specific focus on the top 50 results presented to the user. This approach identifies configurations that prioritize relevant documents effectively and ensures that previously retrieved relevant documents remain prominent in subsequent results.

3 Results

3.1 Use case 1 : RAG AND «code generation»

End-to-end process analysis

Comparison of Configurations Based on the Number of Relevant

The configurations were first compared based on the total number of relevant documents they retrieved throughout the process.
For the query RAG AND «code generation», the original dataset is composed of 24 papers 17 of which are relevant. The final dataset is composed of 264 papers 55 of which are relevant. Therefore in total we have a set of **55 relevant papers**, all others are considered as irrelevant. First we can have a look on table 3.1, which presents the number of relevant papers found by each configuration when considering the end-to-end process :

configuration	nb relevant	total papers
`baseline`	17	24
seed-query_sim-cos	44	6754
seed-query_hyde-sim-cos	44	6754
seed-query_hyde-sim-cos_description	**54**	14544
best-paper_hyde-sim-cos	47	4148
best-paper_hyde-sim-cos_description	**54**	14499

Table 3.1: Number of retrieved papers and number of relevant ones found by each configuration

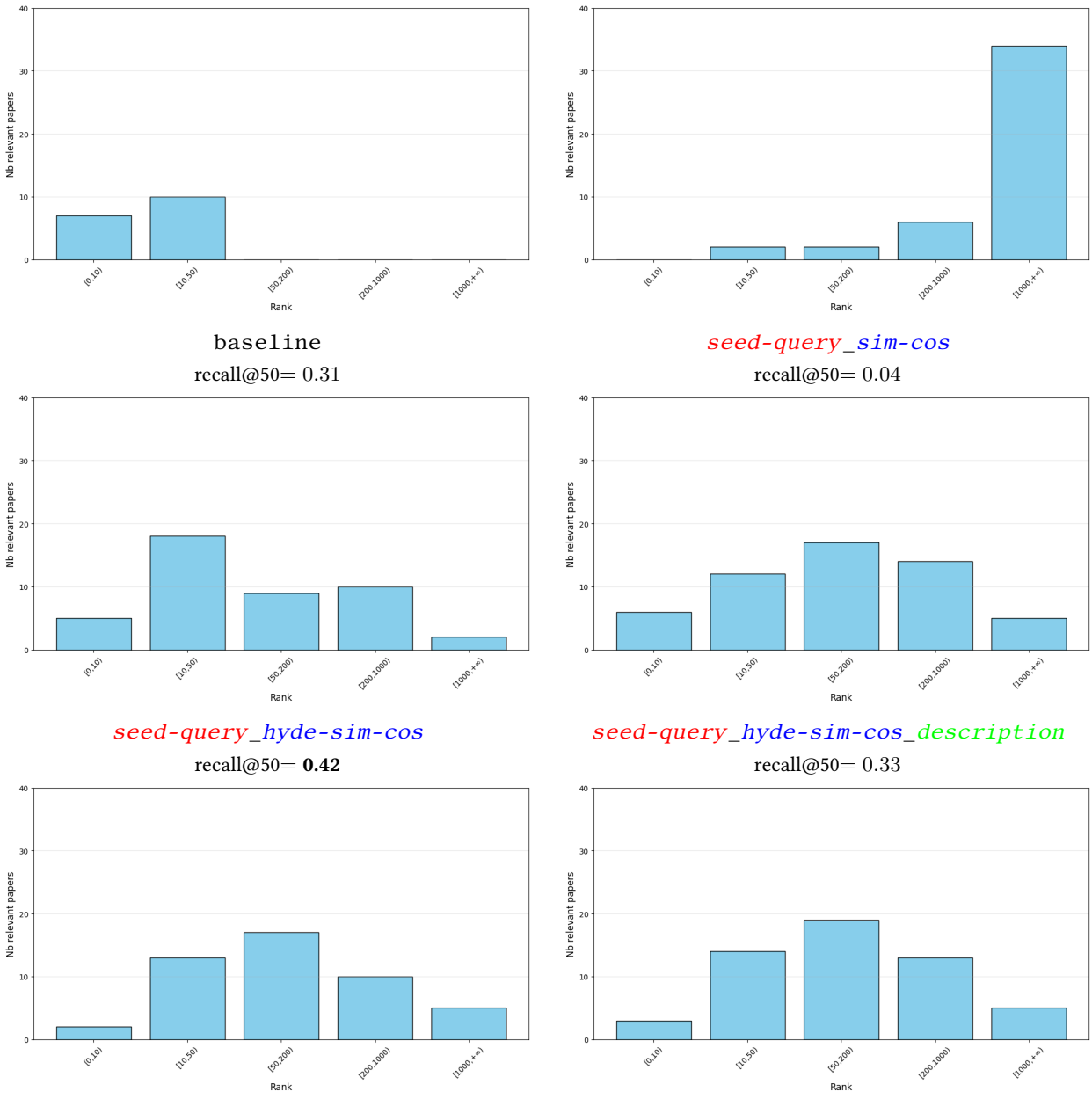
Remember that this number of relevant papers does not correspond to what will be presented to the user. These are all the relevant papers found with the 10 queries whereas afterwards the user will only be given the first 50 papers sorted by score. But it does allow us to observe which configuration finds the most relevant documents in total.
Here both configurations with use of *description* are the best when considering this number but they also retrieved a much larger number of papers in total. That is why we need to look at the rank of relevant papers among them.

Ranking of Relevant Documents by Score

The position of relevant documents within the ranked results was analyzed for each configuration. This evaluation examined whether relevant documents tended to appear among the highest-ranked documents, reflecting the effectiveness of the scoring mechanisms:

- A high concentration of relevant documents in the top ranks indicates effective prioritization.
- Conversely, a dispersed distribution suggests room for improvement in scoring precision.

The figure 3.1 shows the position of relevant papers among all papers found and sorted by score, for each configuration. The recall@50 recall is also included.



`baseline`
recall@50= 0.31

seed-query_sim-cos
recall@50= 0.04

seed-query_hyde-sim-cos
recall@50= **0.42**

seed-query_hyde-sim-cos_description
recall@50= 0.33

best-paper_hyde-sim-cos
recall@50= 0.27

best-paper_hyde-sim-cos_description
recall@50= 0.31

Figure 3.1: Ranking position of relevant papers

- `baseline` serves purely as a comparison point. Since it does not utilize query expansion or any scoring system, it provides no meaningful insights into the effectiveness of focused crawling techniques. Its sole purpose is to establish a reference for the performance of more sophisticated configurations.
- for the configuration *seed-query_sim-cos* most relevant papers are ranked beyond the 1000th position. This ranking is highly unsatisfactory from a user perspective. The goal of a focused crawler is to present relevant documents within the top ranks, as users are assumed in this work to review only the first 50 documents. The poor performance of this configuration indicates that the *sim-cos* scoring system fails to prioritize relevant papers effectively, rendering it unsuitable for practical use in this context.
- Other Configurations: Results from the other configurations show a more nuanced picture. Incorporating the *hyde-sim-cos* scoring system significantly improves the ranking of relevant papers compared to *sim-cos* alone.
  Among all configurations, *seed-query_hyde-sim-cos* achieves the best performance. In this configuration, 23 relevant papers are ranked within the top 50, making it the most promising approach for focused crawling. This outcome suggests that the combination of a well-designed seed query with the more advanced *hyde-sim-cos* scoring system effectively surfaces relevant documents, meeting the practical requirements of ranking and user accessibility.

Overlap and Diversity in Relevant Documents Retrieved

The overlap in relevant documents retrieved by different configurations was analyzed to understand the extent to which configurations complement or replicate each other. This analysis highlights:

- Configurations that retrieve a similar set of relevant documents, suggesting redundancy.
- Configurations that identify unique relevant documents, indicating potential for synergy in combining approaches.

The chart 3.2 called Upset plot visualizes not only which documents each crawler has found but also the intersections among their results (documents found in common). It consists of several components, here's how to read the information :

- Main Histogram (top-right):
  - Each bar represents the size of a group of documents.
  - A group can correspond to a unique set of documents retrieved by a single crawler or an intersection of multiple crawlers (documents retrieved by all in that group).
  - The height of the bar indicates the number of documents in the group.
- Binary Matrix (bottom):
  - Each horizontal line corresponds to a crawler.
  - Dots in each column show which crawlers contribute to the intersection represented by the bar above.
  - A filled dot indicates that the crawler is part of the intersection.
  - Lines connecting multiple dots represent an intersection among those crawlers.
- Individual Set Histogram (left):

– Displays the total number of documents retrieved by each crawler, independently of the others.

This visualization helps evaluate whether the crawlers retrieve many overlapping documents or cover distinct regions of the document space:

- A high number of unique bars indicates that the crawlers are exploring distinct areas.
- Significant intersections suggest redundancy among the crawlers.
- The chart can also highlight dominant intersections, such as when two crawlers consistently find the same documents.

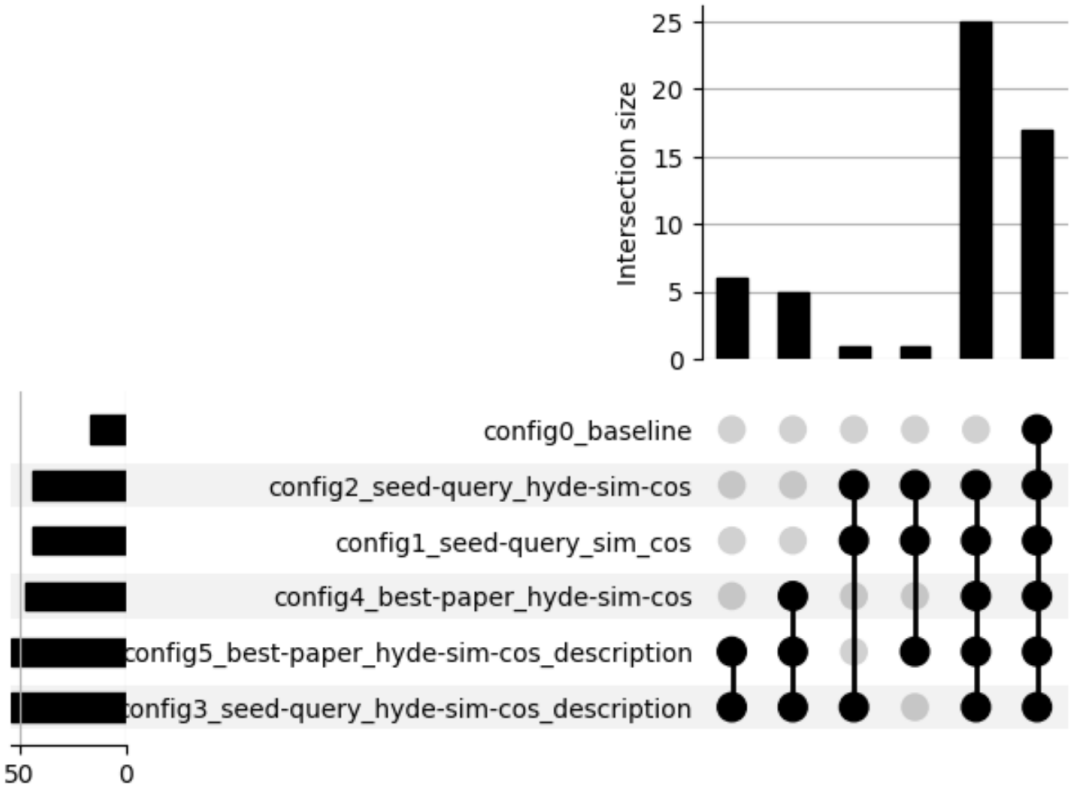


Figure 3.2: Overlap and Diversity in Relevant Documents Retrieved

The 17 relevant documents from the original dataset were successfully retrieved by all configurations. Beyond this, all configurations (excluding the baseline) collectively identified an additional 25 relevant documents. This significant overlap suggests that the shared components or methodologies among these configurations are effective at uncovering a common subset of relevant documents not present in the original dataset. Smaller intersections were observed among other configurations. Notably, there is an overlap of six documents between *seed-query_hyde-sim-cos_description* and *best-paper_hyde-sim-cos_description*, both of which incorporate the *description* of the initial query. This specific overlap implies that leveraging query descriptions can play a meaningful role in identifying certain subsets of relevant documents, even if these subsets are not shared across all configurations.

Query-by-Query Performance Evaluation

So far, we've presented the results obtained at the end of the process, i.e. once all 10 queries have been used. But it's interesting to see how recall and precision evolve with each new query. Perhaps stopping

the process earlier will also improve results.
To measure the progressive improvement of the focused crawler, each query was analyzed individually, focusing on the quality of the top 50 results presented to the user:

- The number of relevant documents among the top 50 results was recorded for each query. This was used to assess whether the system consistently improved its performance in retrieving relevant documents as more queries were issued.
- The behavior of the system was further examined to ensure that relevant documents found in earlier queries remained among the top-ranked results in subsequent queries. For example, if a configuration retrieved a total of 55 relevant documents but failed to present more than 22 in the top 50 for any single query, this indicates that some relevant documents were displaced by less relevant ones in subsequent queries. This behavior suggests inefficiencies in retaining high-quality results across iterations.

This detailed evaluation procedure provides a comprehensive understanding of the strengths and limitations of each configuration, highlighting their ability to uncover relevant documents, maintain precision, and improve user experience over time. Results are presented in the figure 3.3

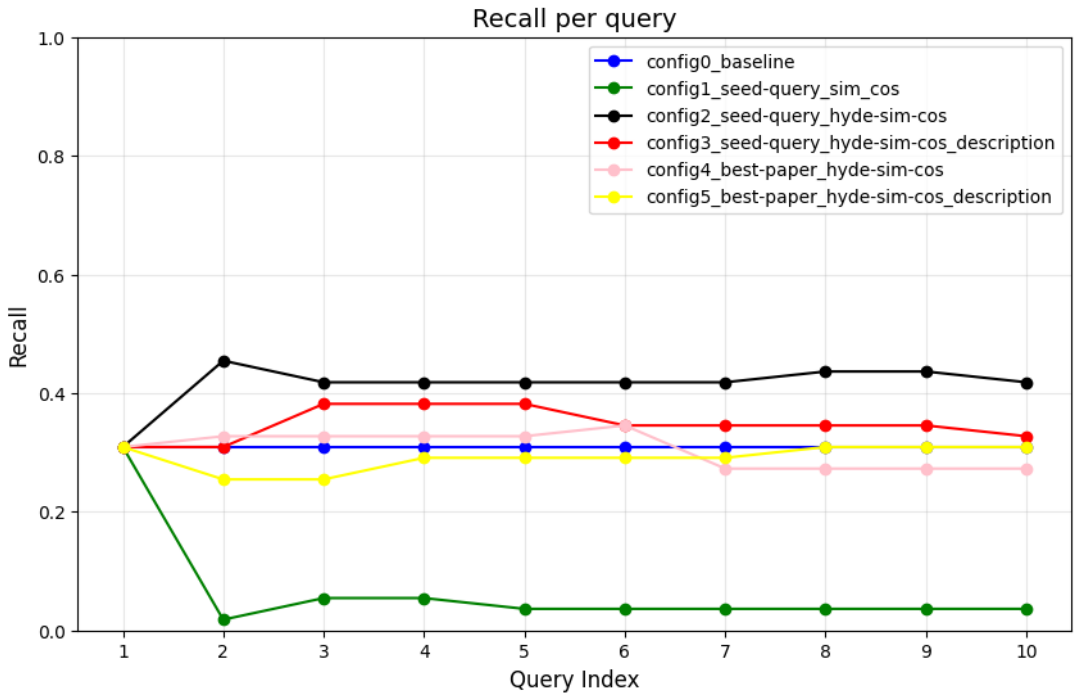


Figure 3.3: Recall by query for each configuration

For `config0_baseline` there is only the initial query. To be abble to compare with others we just repeat the initial query 10 times.

3.2 Use case 2 : "metal-organic frameworks" AND "material design" AND "properties"

End-to-end process analysis

For the query "metal-organic frameworks" AND "material design" AND "properties", the original dataset is composed of 9 papers 6 of which are relevant. The final dataset is composed of 306 papers 50 of

which are relevant. Table 3.1 presents the number of relevant papers found by each configuration when considering the end-to-end process :

Comparison of Configurations Based on the Number of Relevant

configuration	nb relevant papers	total papers
baseline	7	9
seed-query_sim-cos	44	9789
seed-query_hyde-sim-cos	44	9789
seed-query_hyde-sim-cos_description	37	1913
best-paper_hyde-sim-cos	45	11201
best-paper_hyde-sim-cos_description	**49**	7788

Table 3.2: Number of relevant papers found by each configuration

Ranking of Relevant Documents by Score

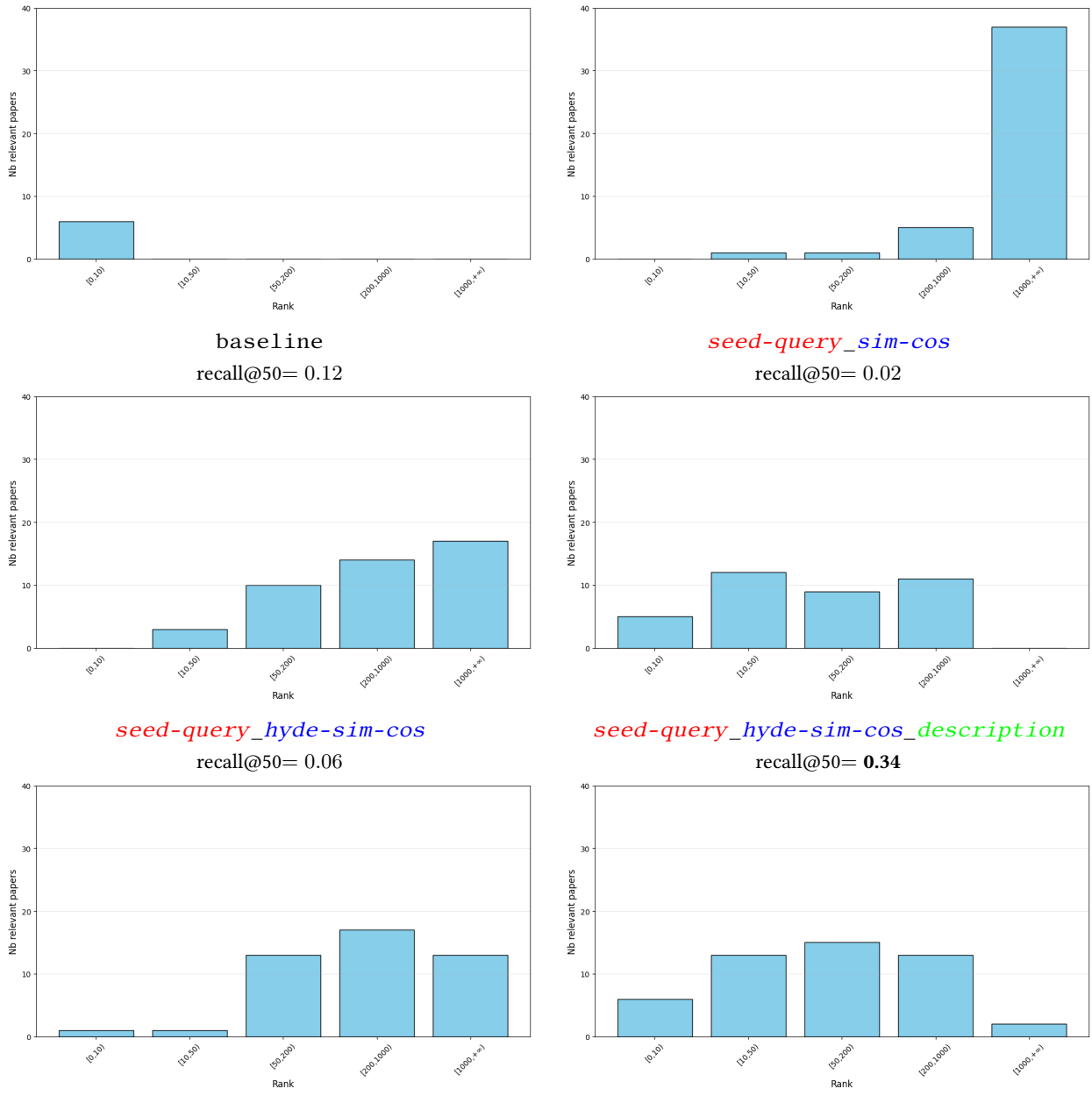


Figure 3.4: Ranking position of relevant papers

Overlap and Diversity in Relevant Documents Retrieved

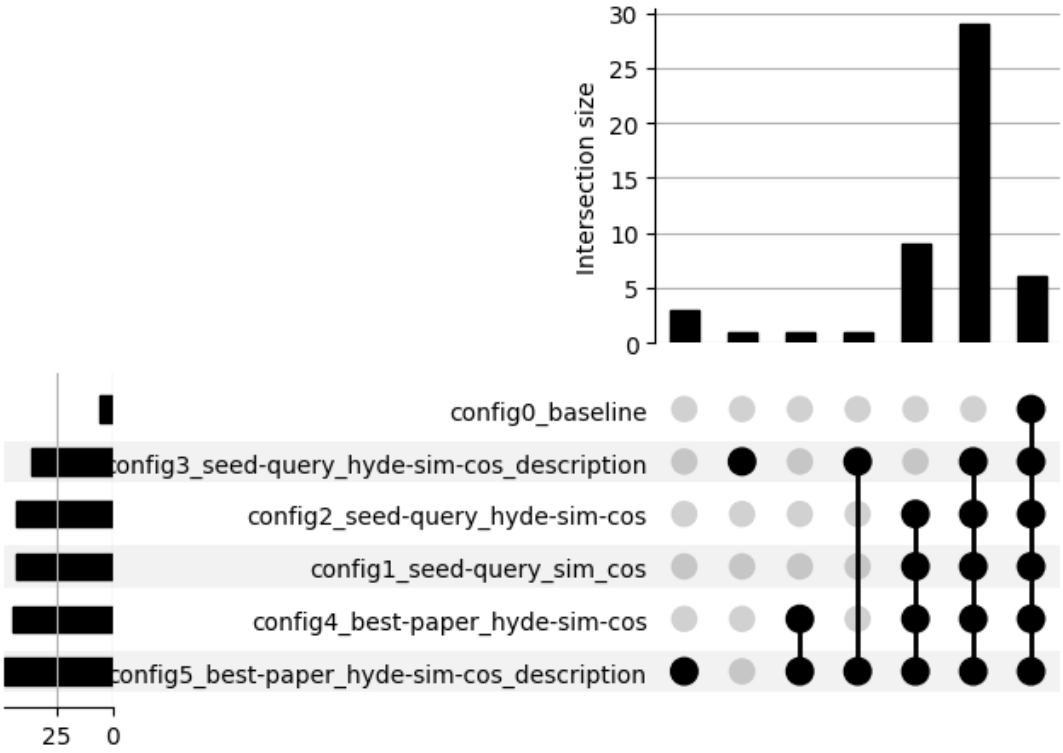


Figure 3.5: Overlap and Diversity in Relevant Documents Retrieved

Query-by-Query Performance Evaluation

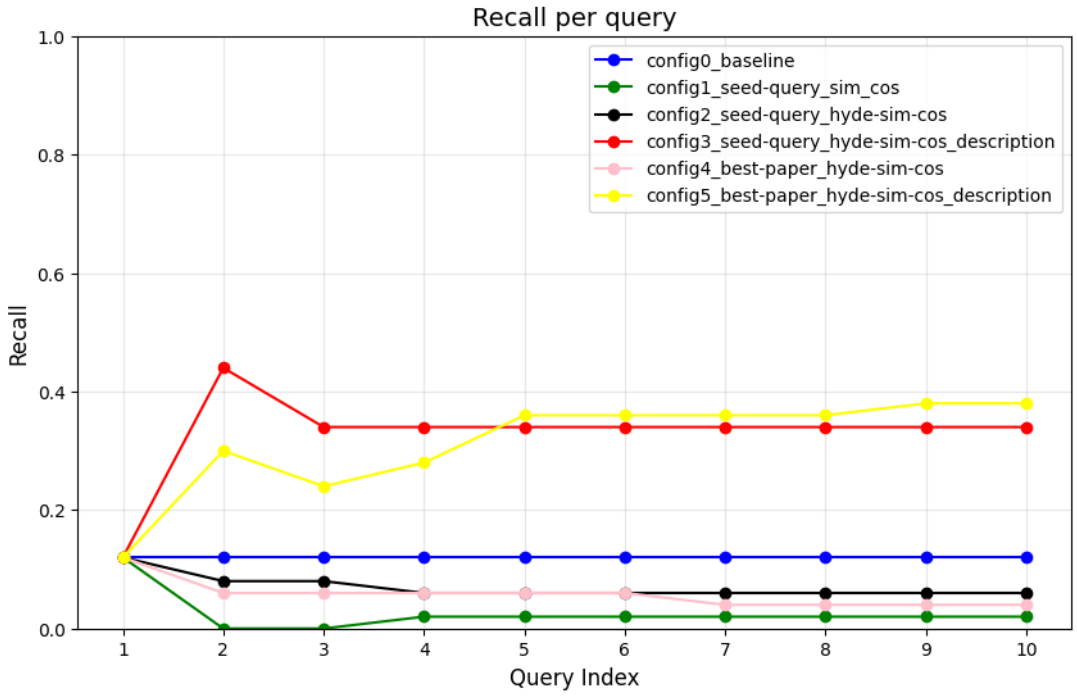


Figure 3.6: Recall by query for each configuration

For this use case, the use of *description* improves results. All other configurations perform worse than the baseline in terms of recall@50, both for the overall process and when looking at query by query.

3.3 Use case 3 : "Machine Learning" AND (diffusion OR diffusivity) AND (MOFs OR ZIFs OR "metal-organic frameworks" OR COFs OR "covalent-organic frameworks)

As a reminder, this use case is one in which many documents are returned and very few are relevant. We therefore need a good scoring system to rank relevant documents as high as possible. We don't study the query expansion part, since we assume that we already have all the documents relevant to the initial query.
Here are the ranks of the 5 relevant papers among all the papers sorted by the score given by the classifiers:

- without classifier : 0, 1, 13, 63, 135
- *sim-cos* : 2285, 3155, 3966, 4054, 5035
- *hyde-sim-cos* : 12, 14, 34, 66, 533
- *hyde-sim-cos_description* : 3, 11, 70, 130, 435

The sim-cos classifier alone is ineffective. It fails to correctly capture the scientific relevance of the papers, likely due to ambiguity or lack of context in the initial query. The HYDE classifiers provide improvements, but their performance is still insufficient to surpass ChemXirv's default ranking. ChemXirv's default ranking is likely based on an algorithm optimized for its specific database. This algorithm takes into account factors such as popularity, citations, or exact search terms, which are invisible to the classifiers used here. Also the poor performance of the classifiers could be linked to the specific vocabulary of these papers, in particular the abbreviation MOF: with a specialized embedding model, we could hope for better results.

3.4 Qualitative Evaluation of Expanded Queries and HYDE

In the appendix, you can find the different queries generated and the HYDE of the configurations using them. The user has rated them according to their relevance.

Overall, queries and HYDE documents generated without incorporating the user-provided description of the initial query are rarely relevant to the domain of interest. When the LLM is unfamiliar with specific terms, it may produce hallucinations, which can subsequently reduce the effectiveness of the focused crawling configuration. For instance, the LLM often misinterprets the term "RAG" (Retrieval-Augmented Generation) as "rapid application generator." However, when a description is provided, both the queries and HYDE documents become significantly more relevant.

Conclusion

The experiments conducted in this work highlight the promising use of LLM-based approaches for focused crawling of scientific papers. In the context of searching for new relevant papers, the query expansion strategy through LLM-generated queries proved effective. It enables users, especially those lacking in-depth knowledge of the domain, to discover papers they might not have identified on their own. This is particularly useful when they are unsure which keywords to include in their queries. In such cases, classifiers based on the HYDE method combined with similarity calculations are sufficient to reduce the number of results presented to the user. Additionally, the inclusion of a complementary description of the query provided by the user further enhances performance, demonstrating significant potential for focused crawling.

Despite the promising results, several limitations and areas for improvement have been identified. In the case of an expert providing a well-constructed and targeted query, the limitations of these classifiers become more apparent. Although the HYDE method offers a significant improvement over traditional similarity calculations, it still falls short of achieving a sufficient level of precision. Furthermore, it is worth noting that performance also varies depending on the API used to retrieve the papers, underscoring the importance of the environment in which the methods are applied.
Regarding generation by an LLM, the prompts used for querying the Large Language Model (LLM) should be refined. The prompt strategy employed in this work could benefit from testing different prompts, possibly incorporating few-shot learning techniques. Specifically, the phrase "expand the query" might not be well-represented in the training data of the LLM, and we made the assumption that the model would understand this concept without prior specific training. Exploring alternative formulations or providing more explicit examples could help improve the query expansion process. Another limitation lies in the use of a generic embeddings model. For better similarity calculations, it would be advantageous to use domain-specific embeddings tailored to the research area of interest. This would likely enhance the precision of the results, as demonstrated in Use Case 3, where the classifier's performance was actually worse than using the search engine without classifiers. A specialized model could offer more accurate semantic matching between documents and queries, particularly for niche topics. Additionally, incorporating active learning techniques could significantly improve the classifier's performance. Active learning could involve gathering feedback from the user at each iteration, allowing them to validate the relevance of documents retrieved and refine queries as needed. This feedback loop could enable the system to adapt and become more effective over time. For instance, users could provide input on the relevance of documents found by a particular query, or specify terms they do not want to appear in subsequent queries. Similarly, in the context of the HYDE method, user validation would allow for more precise generation of hypothetical documents and could ensure that the model's results align better with the user's expectations. Lastly, while the HYDE method was an interesting approach, an alternative strategy could involve calculating the similarity of the papers with seed pages rather than relying on a generated "fake" document. Seed pages—predefined, relevant pages selected by the user—could offer a more grounded and contextually accurate reference for calculating similarity, potentially improving the relevance of the retrieved documents.
By addressing these limitations and integrating these improvements, the focused crawling system could become significantly more effective, tailored to the specific needs of researchers, and capable of delivering more relevant and precise results.

Overall, the use of Large Language Models for focused crawling shows great promise in enhancing the retrieval of relevant scientific papers. While there are still areas for improvement, particularly in classifier precision for expert queries, the combination of query expansion and the HYDE method demonstrates significant potential for improving research efficiency and supporting users with varying levels of domain expertise.

References

[1] B. Chakrabarti, S. ; van den Berg M.; Domc. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Science and Engineering*, 1999.

[2] Freire J. Pham K., Santos A. Bootstrapping domain-specific content discovery on the web. 2019.

[3] A.; Kalpoutzakis E.; Giannakopoulos G. Axiotis, E.; Kontogiannis. A personalized machine-learning-enabled method for efficient research in ethnopharmacology. the case of the southern balkans and the coastal zone of asia minor. *Applied Sciences*, 2021.

[4] Kelesis D. Pollatos V. Paliouras G. ; Kontogiannis, A. and G. Giannakopoulos. Tree-based focused web crawling with reinforcement learning. 2022.

[5] Soares da Silva A. Freire J. Moura E. Vieira K., Barbosa L. Finding seeds to bootstrap focused crawlers. 2015.

[6] Qin Z. Wang X. Bendersky M. Jagerman R., Zhuang H. Query expansion by prompting large language models. 2023.

[7] Wei F. Wang L., Yang N. Query2doc: Query expansion with large language models. 2023.

[8] Lin J. Callan J. Gao L., Xueguang Ma X. Precise zero-shot dense retrieval without relevance labels. 2022.

Appendix

Use Case 1

Initial query q0 : 'RAG AND "code generation"'
Description of initial query used : 'Code generation is an area of artificial intelligence that aims to automate parts of software development. Retrieval-Augmented Generation (RAG) models are a novel approach in this field, combining information retrieval and text generation to produce context-aware code. These methods could help improve the relevance and quality of generated code, making them valuable for a wide range of applications, from prototyping to optimizing software for specific tasks.'

Table 3.3: Relevance of queries - Use Case 1

Configuration	Query	Relevance
config1_ *seed-query* *sim-cos*	Expanded Query 1: RAG (code generation OR code snippet generator OR code template creation)	1
	Expanded Query 2: (RAG OR "Regex pattern matching" OR "Text pattern recognition") AND (code generation OR "algorithmic code creation" OR "automated coding")	0
	Expanded Query 3: (RAG OR "rapid application generation") AND (code generation OR "automated code creation")	0
	Expanded Query 4: (RAG OR 'recompilation' OR 'regeneration') AND (code generation OR 'automated code completion' OR 'intelligent code completion')	0
	Expanded Query 5: (RAG OR "rapid application generator" OR "code template") AND (code generation OR "program code" OR "source code generation")	0
	Expanded Query 6: (RAG OR rapid application generation) AND code generation (OR code *generation* OR code *gen*)	0
	Expanded Query 7: (RAG OR "rapid application generator") AND (code generation OR software generation)	0
	Expanded Query 8: (RAG OR "rapid" OR "agile") AND (code generation OR "software development" OR "generate code")	0
	Expanded Query 9: (RAG OR rapid application generator) AND (code generation OR code snippet generation)	0
config2_ *seed-query* *hyde-sim-cos*	Expanded Query 1: RAG (code generation OR code snippet generator OR code template creation)	1
	Expanded Query 2: (RAG OR "Regex pattern matching" OR "Text pattern recognition") AND (code generation OR "algorithmic code creation" OR "automated coding")	0
	Expanded Query 3: (RAG OR "rapid application generation") AND (code generation OR "automated code creation")	0
	Expanded Query 4: (RAG OR 'recompilation' OR 'regeneration') AND (code generation OR 'automated code completion' OR 'intelligent code completion')	0
	Expanded Query 5: (RAG OR "rapid application generator" OR "code template") AND (code generation OR "program code" OR "source code generation")	0
	Expanded Query 6: (RAG OR rapid application generation) AND code generation (OR code *generation* OR code *gen*)	0

Configuration	Query	Relevance
	Expanded Query 7: (RAG OR rapid application generation) AND (code generation OR software generation)	0
	Expanded Query 8: (RAG OR "rapid" OR "agile") AND (code generation OR "software development" OR "generate code")	0
	Expanded Query 9: (RAG OR rapid application generator) AND (code generation OR code snippet generation)	0
config3_ *seed-query* *hyde-sim-cos* *description*	Expanded Query 1: (RAG OR Retrieval-Augmented Generation) AND (code generation OR code automation OR automatic code completion)	1
	Expanded Query 2: (RAG OR retrieval-augmented generation) AND (code generation OR automated code completion) AND (prototyping OR software optimization)	1
	Expanded Query 3: (RAG OR Retrieval-Augmented Generation) AND (code generation OR code snippet generation OR code completion) AND (artificial intelligence OR machine learning OR AI) AND (software development OR prototyping OR optimization)	1
	Expanded Query 4: (RAG OR Context-Aware Code Generation) AND (code generation OR software development automation) AND (prototyping OR optimization OR specific task optimization)	1
	Expanded Query 5: (RAG OR Retrieval-Augmented Generation) AND (code generation OR code automation OR software development OR artificial intelligence OR prototyping OR optimization)	1
	Expanded Query 6: (RAG OR Generative Model) AND (code generation OR code completion OR auto coding) AND (software development OR prototyping OR optimization)	1
	Expanded Query 7: (RAG OR Retrieval-Augmented Generation) AND (code generation OR automated code creation OR AI-assisted coding)	
	Expanded Query 8: (RAG OR Retrieval-Augmented Generation) AND (code generation OR code completion) AND (artificial intelligence OR machine learning) AND (software development OR software engineering)	1
	Expanded Query 9: (RAG OR Retrieval-Augmented Generation) AND (code generation OR code auto-generation OR code completion OR automated coding) AND (software development OR prototyping OR optimization OR task-specific software)	1
config4_ *best-paper* *hyde-sim-cos*	Expanded Query 1: (rag AND (code generation OR code snippet OR auto generated code))	1
	Expanded Query 2: RAG AND code generation OR (Business Object AND "code generation")	1
	Expanded Query 3: RAG AND (code generation OR code snippet OR auto-generated code)	1
	Expanded Query 4: (RAG AND (code generation OR code generation tool OR automated coding))	1
	Expanded Query 5: (rag OR RAG-related) AND (code OR coding) AND (generation OR code generation OR code generation process)	1
	Expanded Query 6: RAG code generation algorithm or "code generation" software	1
	Expanded Query 7: (rag OR RAG*) AND (code generation OR code generating)	1
	Expanded Query 8: (RAG OR "rapid agile development" OR "rapid application generation") AND (code generation OR "model code" OR "generating models")	0

Configuration	Query	Relevance
	Expanded Query 9: (RAG OR "rapid application generation") AND (code generation OR "automated code creation")	0
config5_ *best-paper* *hyde-sim-cos* *description*	Expanded Query 1: RAG model AND (code generation OR code completion OR text completion) AND (prototyping OR optimization OR task-specific software development)	1
	Expanded Query 2: RAG model AND code generation AND context-aware coding	1
	Expanded Query 3: (RAG OR Retrieval-Augmented Generation) AND (code generation OR code completion OR auto-completion) AND (software development OR prototyping OR optimization)	1
	Expanded Query 4: (RAG OR Retrieval-Augmented Generation) AND (code generation OR automated code completion OR context-aware code)	1
	Expanded Query 5: (RAG OR Retrieval-Augmented Generation) AND code generation AND (prototyping OR optimization OR software development OR artificial intelligence)	1
	Expanded Query 6: (RAG OR Retrieval-Augmented Generation) AND (code generation OR code completion OR auto-completion)	1
	Expanded Query 7: (code generation AND Retrieval-Augmented Generation) OR (context-aware code AND machine learning algorithms for code completion)	1
	Expanded Query 8: (RAG OR Retrieval-Augmented Generation) AND (code generation OR code completion OR code optimization) AND (prototyping OR development OR software optimization)	1
	Expanded Query 9: (RAG OR Model-Aware Code Generation) AND code generation AND (context-aware OR contextual)	1

Table 3.4: Relevance of HYDE - Use Case 1

Configuration	Title	Abstract	Relevance	Cosine Similarity with q_0
seed-query *hyde-sim-cos*	'Rethinking the Role of Recursive Assembly Grammar in Code Generation: A Novel Approach to Automating Software Development'	'This study explores the application of recursive assembly grammar (RAG) in code generation, with a focus on automating software development. The objectives of this research are threefold: (1) to investigate the potential of RAG as a formal specification language for generating efficient and readable code; (2) to develop a novel algorithm for applying RAG in code generation, leveraging insights from machine learning and natural language processing; and (3) to evaluate the efficacy of this approach in terms of productivity gains, code quality, and maintainability. The proposed methodology involves a hybrid approach combining symbolic manipulation with gradient-based optimization techniques. Our results indicate that the novel algorithm outperforms existing code generators in terms of code quality and readability, while also achieving significant reductions in development time. The implications of this research are far-reaching, suggesting new avenues for improving software productivity, reducing bugs, and enhancing collaboration among developers.'	0	0.863

Continued on next page

Configuration	Title	Abstract	Relevance	Cosine Similarity with q_0
seed-query *hyde-sim-cos* *description*	'Improving Code Generation through Retrieval-Augmented Generation: A Novel Approach for Enhanced Context-Aware Software Development'	'This research paper explores the application of Retrieval-Augmented Generation (RAG) models in code generation, a field of artificial intelligence aiming to automate parts of software development. By combining information retrieval and text generation, RAG models produce context-aware code that can improve the relevance and quality of generated code. This study investigates the objectives, methodology, and implications of integrating RAG models into existing code generation frameworks. The proposed approach leverages pre-trained language models and knowledge graphs to retrieve relevant code snippets and generate high-quality, context-specific code. Our results demonstrate a significant improvement in code completion accuracy and quality compared to traditional approaches. The findings have far-reaching implications for the software development industry, enabling the creation of more efficient, scalable, and maintainable software systems.'	1	0.868
best-paper *hyde-sim-cos*	'Automating Software Development: A Study on Rapid Application Generation (RAG) for Code Generation'	"This study explores the potential of Rapid Application Generation (RAG) in automating software development, with a specific focus on code generation. The objective is to investigate the feasibility and effectiveness of using RAG tools to generate high-quality code. A mixed-methods approach was employed, combining both qualitative and quantitative data collection methods. A survey of 100 software developers was conducted to gather insights on their experiences with RAG tools, while a comparative analysis of open-source projects utilizing RAG was performed to evaluate the generated code's quality. The results indicate that RAG can significantly reduce development time and improve code maintainability. However, challenges related to customizability and error handling are still prevalent. The study contributes to the existing body of research by highlighting the potential benefits and limitations of RAG in software development. The findings have implications for both researchers and practitioners seeking to automate software development using RAG tools."	0	0.873
best-paper *hyde-sim-cos* *description*	'Retrieval-Augmented Code Generation: A Novel Approach for Improving Context-Aware Software Development'	'This research paper explores the application of Retrieval-Augmented Generation (RAG) models in code generation, a key area of artificial intelligence. By combining information retrieval and text generation capabilities, RAG models can produce context-aware code that is more relevant and of higher quality compared to existing methods. The objectives of this study are threefold: (1) to investigate the feasibility of RAG models for code generation, (2) to evaluate their performance in producing high-quality code, and (3) to explore the potential applications of RAG-based code generation across various software development tasks. The methodology employed in this research involves designing and training RAG models using large-scale code datasets and benchmarking them against existing code generation techniques. The results indicate that RAG models can significantly improve the quality and relevance of generated code, making them a valuable tool for developers. The implications of this research are far-reaching, with potential applications in software prototyping, optimization, and maintenance, among others.'	1	0.873

Use Case 2

Initial query q0 : '"metal-organic frameworks" AND "material design" AND "properties"'
Description of initial query used : 'Metal-organic frameworks is a family of nanoporous materials whose structure can be modified on the molecular level, towards tailoring their macroscopic properties.

Property can be, for example, the amount of a specific gas species adsorbed in the pores of the material. However the correlation between structure and property value is complicated. Using ML predictive models (structure as input, and property values as output) is useful, but still is burdened by a trial-and-error approach. It would be useful to have an AI system that takes as input the desired property value and outputs the structure, which is the exact opposite direction of traditional ML models.'

Table 3.5: Relevance of queries - Use Case 2

Configuration	Query	Relevance
seed-query *sim-cos*	Expanded Query 1: (metal-organic frameworks (design $ properties) AND material design (frameworks $ properties) AND porous materials (frameworks $ properties))	
	Expanded Query 2: (metal-organic frameworks) AND (material design) AND (properties) AND (porosity) AND (surface area) AND (adsorption) AND (desorption) AND (gas storage) AND (catalysis)	
	Expanded Query 3: (metal-organic frameworks AND material design) AND (properties OR thermal insulation OR gas adsorption OR mechanical properties OR optical properties OR electrical conductivity OR catalytic properties)	
	Expanded Query 4: ("metal-organic frameworks" OR "MOFs") AND ("material design" OR "hierarchical materials" OR "porous materials") AND ("properties" OR "thermodynamics" OR "sorption properties")	
	Expanded Query 5: (	
	Expanded Query 6: (metal-organic frameworks OR MOFs) AND (material design OR adaptive materials) AND (properties OR characteristics OR functionality)	
	Expanded Query 7: ("Metal-Organic Frameworks" OR "MOFs") AND ("Material Design" OR "Design Principles" OR "Crystalline Materials") AND ("Properties" OR "Thermodynamics" OR "Optoelectronics" OR "Sorption Capacity")	
	Expanded Query 8: ("metallorganicframeworks" OR "MOFs") AND ("materialdesign" OR "selfassemblingmaterials" OR "mesoscale engineering") AND ("properties" OR "thermodynamics" OR "electricalconductivity")	
	Expanded Query 9: (('metal-organic frameworks' OR 'MOFs') AND ('material design' OR 'self-assembled structures') AND ('properties' OR 'functional properties'))	
seed-query *hyde-sim-cos*	Expanded Query 1: (metal-organic frameworks (design $ properties) AND material design (frameworks $ properties) AND porous materials (frameworks $ properties))	
	Expanded Query 2: (metal-organic frameworks) AND (material design) AND (properties) AND (porosity) AND (surface area) AND (adsorption) AND (desorption) AND (gas storage) AND (catalysis)	
	Expanded Query 3: (metal-organic frameworks AND material design) AND (properties OR thermal insulation OR gas adsorption OR mechanical properties OR optical properties OR electrical conductivity OR catalytic properties)	
	Expanded Query 4: ("metal-organic frameworks" OR "MOFs") AND ("material design" OR "hierarchical materials" OR "porous materials") AND ("properties" OR "thermodynamics" OR "sorption properties")	
	Expanded Query 5: (	
	Expanded Query 6: (metal-organic frameworks OR MOFs) AND (material design OR adaptive materials) AND (properties OR characteristics OR functionality)	

Configuration	Query	Relevance
	Expanded Query 7: ("Metal-Organic Frameworks" OR "MOFs") AND ("Material Design" OR "Design Principles" OR "Crystalline Materials") AND ("Properties" OR "Thermodynamics" OR "Optoelectronics" OR "Sorption Capacity")	
	Expanded Query 8: ("metallorganicframeworks" OR "MOFs") AND ("materialdesign" OR "selfassemblingmaterials" OR "mesoscale engineering") AND ("properties" OR "thermodynamics" OR "electricalconductivity")	
	Expanded Query 9: (('metal-organic frameworks' OR 'MOFs') AND ('material design' OR 'self-assembled structures') AND ('properties' OR 'functional properties'))	
seed-query *hyde-sim-cos* *description*	Expanded Query 1: metal-organic frameworks AND material design AND properties AND predictive modeling AND inverse machine learning OR inverse regression;metal-organic frameworks AND (material design OR material science) AND (properties OR chemical properties OR gas adsorption OR molecular structure) AND (inverse machine learning OR inverse predictive modeling OR inverse predictive regression)	
	Expanded Query 2: metal-organic frameworks AND material design AND properties AND predictive modeling AND inverse machine learning OR inverse regression;metal-organic frameworks AND (material design OR material science) AND (properties OR chemical properties OR gas adsorption OR molecular structure) AND (inverse machine learning OR inverse predictive modeling OR inverse predictive regression)	
	Expanded Query 3: metal-organic frameworks AND material design AND properties AND nanoporous materials AND structure-property correlation AND machine learning predictive models AND inverse predictivity AND structure prediction AND desired property value	
	Expanded Query 4: metal-organic frameworks AND material design AND properties AND nanoporous materials AND structure-property correlation AND machine learning predictive models AND inverse topology optimization	
	Expanded Query 5: metal-organic frameworks AND material design AND properties AND gas adsorption AND nanoporous materials AND molecular modification AND macroscopic property tailoring AND inverse machine learning AND structure prediction AND desired property value input AND predictive models	
	Expanded Query 6: metal-organic frameworks AND material design AND properties AND nanostructured materials AND adsorption AND gas species AND machine learning predictive models AND structure-property prediction AND inverse modeling AND optimization algorithms AND direct synthesis	
	Expanded Query 7: metal-organic frameworks AND material design AND properties AND nanoporous materials AND structure-property correlation AND machine learning predictive models AND inverse problem solving	
	Expanded Query 8: metal-organic frameworks AND material design AND properties AND gas adsorption AND porous materials AND nanoporous materials AND predictive modeling AND inverse machine learning AND structure prediction AND material synthesis AND computational materials science	
	Expanded Query 9: metal-organic frameworks AND material design AND properties AND computational model development AND inverse machine learning AND structure prediction AND materials science AND optimization algorithms AND predictive modeling AND inverse machine learning models	

Configuration	Query	Relevance
best-paper *hyde-sim-cos*	Expanded Query 1: (metal-organic frameworks OR porous materials) AND (material design OR architecture) AND (properties OR mechanical properties OR thermal properties)	
	Expanded Query 2: (metal-organic frameworks) AND (material design) AND (properties) AND (porosity) AND (surface area) AND (adsorption)	
	Expanded Query 3: (metal-organic frameworks) AND (material design) AND properties AND (porosity AND surface area OR adsorption capacity)	
	Expanded Query 4:(metal-organic frameworks AND material design) (AND properties OR functional properties OR structural properties OR thermal properties OR mechanical properties OR optical properties)	
	Expanded Query 5: metal-organic frameworks AND material design properties AND porous structures AND tunable conductivity	
	Expanded Query 6: (metal-organic frameworks$ OR MOFs) AND (material design$ OR smart materials) AND (properties$ OR properties of MOFs)	
	Expanded Query 7: (metal-organic frameworks AND (material design OR material science)) AND properties	
	Expanded Query 8: (metal-organic frameworks AND (material design OR sustainable materials)) AND properties	
	Expanded Query 9: metal-organic frameworks AND material design AND properties AND (structural OR thermal OR optical)	
best-paper *hyde-sim-cos* *description*	Expanded Query 1: metal-organic frameworks AND material design AND properties AND predictive modeling AND inverse machine learning AND structure prediction	
	Expanded Query 2: (metal-organic frameworks AND material design) (structure AND property OR properties OR adsorption OR gas species adsorption) NOT ((model OR AI system OR predictive model) AND (structure AS input AND property values AS output))	
	Expanded Query 3: metal-organic frameworks AND material design AND properties AND predictive modeling AND inverse problem solving	
	Expanded Query 4: (metal-organic frameworks OR MOFs) AND (material design FOR porous structures) AND (properties OR properties optimization) AND (reverse prediction OR inverse property-to-structure prediction) AND (machine learning FOR material design optimization)	
	Expanded Query 5: (metal-organic frameworks OR MOFs) AND (material design OR structural design) AND (properties OR material properties OR macroscopic properties) AND (predictive modeling OR machine learning predictive models) AND (reverse engineering OR inverse design)	
	Expanded Query 6: metal-organic frameworks AND material design AND properties AND (reverse predictive modeling OR inverse property prediction)	
	Expanded Query 7: metal-organic frameworks AND material design AND properties AND inverse predictive modeling	
	Expanded Query 8: (metal-organic frameworks AND (material design OR structural optimization) AND (gas adsorption properties OR predictive modeling))	
	Expanded Query 9: metal-organic frameworks AND material design AND properties AND inverse machine learning OR predictive modeling OF structure FROM desired property values	

Table 3.6: Relevance of HYDE - Use Case 2

Configuration	Title	Abstract	Relevance	Cosine Similarity with q_0
seed-query *hyde-sim-cos*	'Exploring the Properties and Design Principles of Metal-Organic Frameworks in Material Science'	'This study investigates the role of metal-organic frameworks (MOFs) in material design, with a focus on their unique properties and applications. MOFs have gained significant attention in recent years due to their exceptional porosity, thermal stability, and tunable chemical functionality. This research aims to provide an overview of the current state-of-the-art in MOF materials science, highlighting their design principles, synthesis methods, and properties. The objectives of this study are to: (1) review the current literature on MOFs and their applications; (2) develop a novel synthesis method for MOFs with tailored properties; and (3) investigate the mechanical, thermal, and chemical properties of synthesized MOFs. The findings of this research will provide insights into the design principles and properties of MOFs, which can be used to inform the development of new materials for various applications, including energy storage, catalysis, and drug delivery.'		0.874
seed-query *hyde-sim-cos* *description*	'Inverting Machine Learning for Metal-Organic Frameworks: A Deep Learning Approach to Predicting Desired Properties'	'Metal-organic frameworks (MOFs) are nanoporous materials with tailored macroscopic properties, but the correlation between structure and property values is complex. Traditional machine learning (ML) models use structure as input and property values as output, limiting their ability to reverse-engineer desired properties. This paper proposes an AI system that inverts this paradigm by taking desired property values as input and generating optimal MOF structures as output. Leveraging deep learning techniques, our approach learns to predict the relationships between molecular structure and macroscopic properties. We demonstrate the effectiveness of our method using a range of MOFs and test sets, achieving state-of-the-art performance in predicting specific gas adsorption properties. Our work opens up new avenues for the design of MOFs with tailored properties, enabling more efficient and sustainable applications in fields such as energy storage and catalysis.'		0.859

Continued on next page

Configuration	Title	Abstract	Relevance	Cosine Similarity with q_0
best-paper *hyde-sim-cos*	'Unlocking Material Properties: Exploring the Role of Metal-Organic Frameworks in Novel Material Design'	'This study investigates the intersection of metal-organic frameworks (MOFs) and material design, with a focus on understanding their properties and potential applications. MOFs are two-dimensional materials composed of metal centers linked to organic ligands, offering unique structural and chemical properties that make them attractive for various technological applications.The objectives of this research are threefold: (1) to examine the relationship between MOF structure and its physical properties, such as surface area, pore size distribution, and thermal stability; (2) to explore the synthesis methods and techniques used to modify MOF properties, with a focus on material design strategies; and (3) to investigate the implications of these findings for the development of new materials with tailored properties.The methodology employed in this study involves a comprehensive review of existing literature on MOFs and their properties, as well as theoretical modeling and simulation studies. Theoretical models were used to simulate the structural changes that occur during MOF synthesis and modification, while experimental data from existing research was used to validate these predictions.The results of this study demonstrate the potential of MOFs as a platform for material design, with applications in fields such as energy storage, catalysis, and sensing. The findings highlight the importance of understanding the relationship between MOF structure and properties, and provide insights into the synthesis methods and techniques used to modify MOF properties.The implications of this research are significant, as they have the potential to impact various industries, including energy, chemicals, and healthcare. By providing a deeper understanding of MOFs and their properties, this study contributes to the development of new materials with tailored properties, which can be used to address pressing global challenges."'Please note that this is just one possible title and abstract based on the search query, and there are many other possible combinations and interpretations.'		0.869
best-paper *hyde-sim-cos* *description*	'Reversing the Paradigm: Predictive Modeling of Metal-Organic Frameworks from Desired Properties to Structure'	'Metal-organic frameworks (MOFs) are nanoporous materials whose structure can be modified on a molecular level to tailor their macroscopic properties. However, the correlation between structure and property value is complicated. Traditional machine learning (ML) models often rely on trial-and-error approaches by inputting structure as input and outputting property values. This paper proposes an AI system that inverts this paradigm, taking desired property values as input and generating optimal structures as output. The objectives of this research are to develop a predictive model that can accurately forecast MOF properties from their desired values, and to investigate the implications of such an approach for materials design and discovery. The methodology employed includes the development of a novel inverse ML model using meta-learning techniques, which learns to generate structures that correspond to specified property values. The results show promising predictions of MOF properties with high accuracy, demonstrating the potential of this approach for designing new materials with tailored properties.'		0.866

Use Case 3

Initial query q0 : "'"Machine Learning" AND (diffusion OR diffusivity) AND (MOFs OR ZIFs OR "metal-organic frameworks" OR COFs OR "covalent-organic frameworks)"'
Description of initial query used : 'Metal-organic frameworks (MOFs), ZIFs and covalent-organic frameworks (COFs) are families of nanoporous materials whose structure can be modified on the molec-

ular level, towards tailoring their macroscopic properties. However the correlation between structure and property value is complicated. Property can be, for example, the amount of a specific gas species adsorbed in the pores of the material. Machine Learning can be useful in getting data, be trained on them, and extract the aforementioned structure-property correlations.'

Table 3.7: Relevance of HYDE - Use Case 3

Classifier	**Title**	**Abstract**	**Relevance**	**Cosine Similarity with q_0**
hyde-sim-cos	'Enhancing Diffusion in Metal-Organic Frameworks through Machine Learning Optimization'	'The development of efficient gas adsorption and diffusion mechanisms in metal-organic frameworks (MOFs) is crucial for various applications, including gas storage, separation, and catalysis. In this study, we investigate the application of machine learning algorithms to optimize diffusion properties in MOFs. By combining computational simulations with machine learning techniques, we aim to identify the most effective molecular structures and surface modifications that enhance diffusion rates. Our results show significant improvements in diffusion coefficients for a range of MOF materials, leading to enhanced performance in gas storage and separation applications. The work demonstrates the potential of machine learning as a tool for rational design of MOFs with optimized diffusion properties, paving the way for next-generation MOF-based technologies.'		0.897
hyde-sim-cos *description*	'Unraveling Structure-Property Correlations in Metal-Organic Frameworks (MOFs) using Machine Learning: A Review of ZIFs, COFs, and Generalized Diffusion Models'	'The relationship between the structure and properties of metal-organic frameworks (MOFs), zeolime-inspired frameworks (ZIFs), and covalent-organic frameworks (COFs) remains complex. The unique ability of these materials to be tailored at the molecular level raises questions about their macroscopic behavior. This review aims to explore the application of machine learning (ML) in extracting structure-property correlations from MOFs, ZIFs, and COFs. We discuss various machine learning techniques that can be employed to analyze the data generated by these frameworks, including diffusion models that account for diffusivity. Our goal is to provide a comprehensive overview of the current state of research on this topic, highlighting the benefits and limitations of using ML in understanding the intricate relationships between structure and properties in MOFs, ZIFs, and COFs.'		0.916