Benchmarking the Undermind Search Assistant

Thomas Hartke and Joshua Ramette
Undermind.ai

January 5, 2024

Abstract

We outline an end-to-end solution for searching academic literature. This system, Undermind, uses language models as a reasoning engine and classifier at key steps within a structured search process. We benchmark Undermind's performance compared to Google Scholar, showing drastic improvements including a 10× higher concentration of truly relevant results within the top hits. Undermind misses virtually no highly relevant works found by Google Scholar, and in addition returns 10× the total number of relevant results for the median user-generated query.

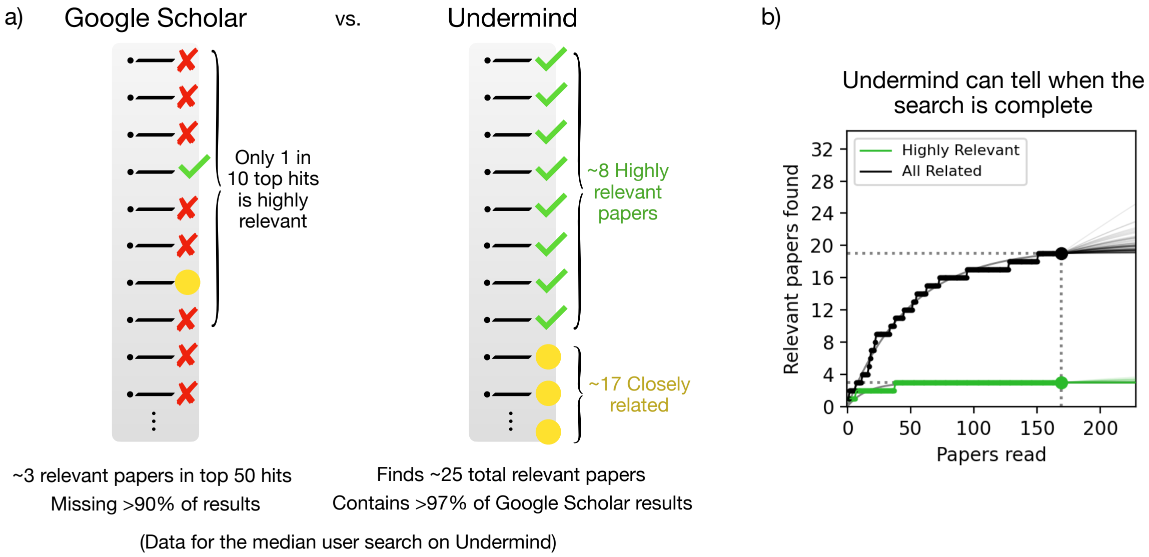

Figure 1: **Undermind is accurate and comprehensive.** We systematically compared the quality of search results returned by Google Scholar and Undermind for ∼300 user-generated queries. For the median user query, shown quantitatively in (a), Undermind discovers 10× more relevant results compared to the top 50 hits on Google Scholar. In addition, Undermind clearly flags (and explains) which results are most relevant at the top of the report. In contrast, on Google Scholar the few relevant hits are tedious to pick out, as they are scattered among mostly irrelevant results (only 1 in 10 are highly relevant). Moreover, Undermind is comprehensive, missing $< 3\%$ of Google Scholar's highly relevant results. See Section 2 for quantitative details. (b) Undermind's rate of finding relevant papers decays exponentially as it explores the database. Powerfully, by tracking this exponential, Undermind can determine how far it must look to ensure it returns all relevant papers.

1 Building an ideal search engine for scientists

A perfect search system should be like a human assistant. It should understand your complex search goals, and then carefully and systematically search an entire literature database to find all precisely relevant results for you. It should explain these results, individually and in aggregate, in a comprehensive report.

Undermind achieves these goals nearly perfectly. It currently accesses the scientific literature database ArXiv,[1], searching within the full texts of 2.3 million papers. It uses a language model (GPT-4) as a reasoning engine at key steps in a structured exploration process. Its search algorithm mimics that of a human, adapting and following citation trails to uncover important papers and reflecting on progress so far to decide next steps. Ultimately, Undermind delivers a precise set of final results exactly relevant to the user's complex search topic, explaining each result in detail. The quality of this report far exceeds that of existing search engines (see Fig. 1 and Fig. 2).

1.1 How it works

There are four steps to Undermind's algorithm:

1. **Basic search:** We identify promising candidate papers using a custom algorithm that combines semantic vector embeddings, citations, and language model reasoning.

2. **Relevance classification:** Given your search query, a high quality language model (GPT-4) accurately classifies each candidate paper based on its full text into 3 categories: highly relevant, closely related (meaning relevant, but slightly off-topic), or ignorable. See Appendix 3.2 for classification accuracy statistics.[2]

3. **Adaptation and exploration:** The algorithm adapts and searches again based on the relevant content it has discovered. This adaptation, which mimics a human's discovery process, makes it possible to uncover every relevant result.

4. **Estimating comprehensiveness:** Undermind tracks how frequently it discovers relevant papers during each search. Undermind initially finds many relevant results, but over time diminishing returns set in, empirically leading to "discovery curves" which are exponential in form (see Fig. 1(b)). Modeling this process allows us to determine when Undermind has found nearly all the relevant works.

2 Key metrics compared to Google Scholar

Here we provide quantitative benchmarks of the quality of Undermind compared to Google Scholar, an academic literature search engine which is often considered the gold standard by researchers. To make this benchmark, we gathered and analyzed the results of ~ 300 user queries submitted to Undermind by scientists in late 2023.

The main results are presented in Fig. 2, showing the number of relevant papers Undermind autonomously returns compared to the number a human could find with reasonable effort using Google Scholar.[3] As a proxy for reasonable human effort on Google, we gather the first page of Google Scholar results for 5 separate keyword searches that re-phrase the user's original query. For a fair comparison, we evaluate the relevance of these Google Scholar results using the same relevance classification subroutine that Undermind employs. Appendix 3.3 describes our methodology and how we translate complex Undermind queries into reasonable keyword search phrases for Google Scholar, and Fig. 4 and Fig. 5 contain further data about the classified results.

2.1 Quantitative results

Undermind quantitatively outperforms Google Scholar in 3 ways:

1. 10× **more relevant results on Undermind vs. the first 5 pages of Google Scholar.** In many cases Google Scholar finds 0 results, while Undermind finds 10-20. Even for searches where Google Scholar returns a few relevant papers, Undermind still returns significantly more. The full distribution of relative performance is shown in Fig. 2.

[1] https://www.arxiv.org/

[2] With accuracy ~98%, Undermind never classifies a highly relevant paper as irrelevant, or an irrelevant paper as highly relevant.

[3] Undermind's classifier was used to identify relevant ArXiv papers in Google Scholar's top 50 results, which typically contained ~30 ArXiv papers.

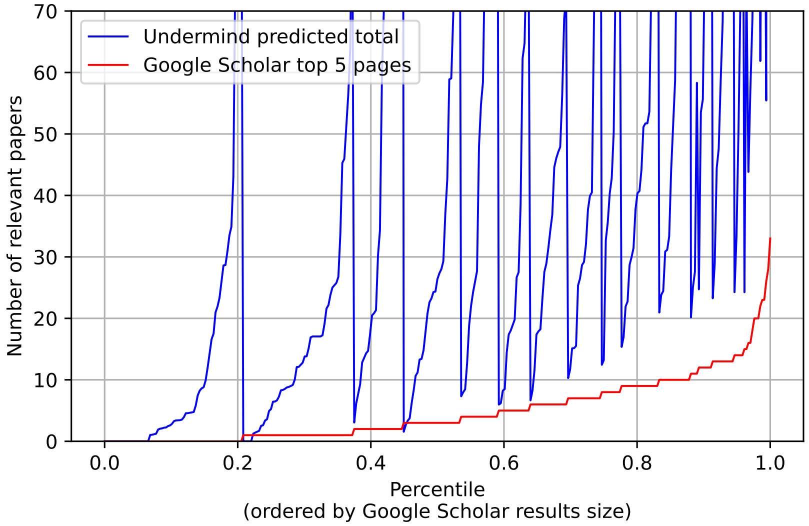

Figure 2: **Undermind finds far more relevant papers than Google Scholar.** Data for 300 user-generated queries. Blue line: The number of relevant results found by converged Undermind searches (or, if not converged, the estimated total findable by Undermind with modest extension). Red line: The number of relevant results found in Google Scholar's top 5 pages for the same queries. The queries are ordered by the number of relevant papers found by Google Scholar, and then further by the number Undermind found. For many queries, Undermind finds 10s of papers while Google Scholar finds nothing (percentile ~ 0.15). For searches with many Google Scholar results, Undermind still finds 3-5× more results (percentile ∼0.9).

2. 10× **higher density of relevant information on Undermind.** Undermind flags relevant works (typically $10 - 20$), explains *why* they are relevant to the user's specific query and goals, and moves them right to the top of each report, as shown in Fig. 1. Each of these highlighted papers is relevant to the topic, with probability greater than 92% (see Table 2). In contrast, when reading Google Scholar, typically only 1 in 10 top hits is highly relevant,[4] requiring an enormous amount of effort to manually parse and filter irrelevant hits. See Fig. 4 for the full distribution.

3. **Exhaustiveness of Undermind:** Once an Undermind search has converged, virtually every highly relevant paper found by Google Scholar is already found by Undermind, within statistical certainty (greater than 97% chance, see Appendix 3.5). This strongly suggests that Undermind finds **nearly all papers that exist on a topic** once a search has converged.[5]

   - Moreover, due to the efficiency of Undermind's search algorithm, reaching convergence is easy for nearly all searches. A typical search already retrieves more than $\sim 85\%$ of results within the first 150 papers explored (see Fig. 3(b)). When desired, the rest can be retrieved by extending a search.

2.2 Further advantages

Further advantages of Undermind, which are harder to quantify, include:

1. **The ability to handle very complex searches:** Because Undermind reads deep within the full texts of papers with the user's exact goals in mind, it can understand and evaluate very complex

[4] This analysis checked the top 10 ArXiv papers returned by Google Scholar.

[5] To see why this is true, suppose that Undermind were **not** exhaustive, and that it only finds a fraction α of some hypothetical larger set of truly relevant works T. Suppose also that Google Scholar and Undermind draw their discovered relevant works mostly independently from T, which is a mild assumption given that Google Scholar and Undermind are completely different algorithms internally. Of the relevant results that Google Scholar finds, a fraction $\sim \alpha$ would be found by Undermind as well, and a substantial fraction $\sim 1 - \alpha$ would be newly discovered relevant papers. However, we've checked that virtually every highly relevant paper found by Google Scholar is also found by a converged Undermind search (see Appendix 3.5 and Fig. 5). This is most consistent with Undermind being truly exhaustive, i.e. $\alpha \approx 1$.

search goals (see Appendix 3.1 for examples of very complex searches submitted by scientists). In contrast, for many user requests, Google Scholar completely fails to return relevant results. This is likely because it is impossible to translate many complex, real world needs and requests into efficient keyword searches.

2. **Knowing how much prior work has been done on a topic** Because of the predictable exponential form of Undermind's discovery process, we can estimate how many relevant works exist on a given topic after initially exploring the database. This gives the user an immediate snapshot of how novel their search topic is, a capability strictly absent from conventional keyword search.

3. **Confirming nothing exists on a topic** Because Undermind is likely truly exhaustive, if Undermind provides no relevant results, one can be reasonably certain nothing exists on the topic. In contrast, if one uses Google Scholar and finds no results, it's impossible to know whether nothing exists, or whether keyword searching with Google Scholar has simply failed (see Fig. 2, left side).

3 Appendix

3.1 Distribution of real user searches on Undermind

User-submitted requests to Undermind vary in complexity and difficulty. However, for each search, the discovery rate of relevant papers follows an exponential form, and saturates after Undermind has found most relevant results, as shown in Fig. 1(b). The variation in the complexity of searches submitted by users causes the time constant as well as the total number of relevant papers found to vary widely between searches.[6]

To convey this variation, in Fig. 3 we show the predicted number of relevant papers and convergence rate for the user searches submitted to Undermind and analyzed in this report. A median search has 24 relevant papers and converges with a time constant of 80 papers evaluated, meaning that a typical Undermind report evaluating 150 papers would immediately find $\sim 85\%$ of all relevant results. Extending this search to read 150 additional papers (300 total) would find $\sim 98\%$ of all relevant results.

To clarify the range of complexities for user queries, we provide a few examples (modified slightly for privacy):

1. Examples of simpler queries

> Topic: *Review articles on quantum computing with Rydberg atoms*
> Additional context: *I want to learn about the general state of the field of quantum computing with Rydberg atoms or arrays of Rydberg atoms.*

> Topic: *Routing trapped ions in a quantum computer*
> Additional context: *Trapped ions are usually routed (moved around, or shuttled) to bring ions closer together to interact and perform quantum gates.*

2. Examples of more complex queries

> Topic: *Tokenization-free large language model architectures, in particular any character-level models which have been shown to achieve compute/accuracy tradeoffs comparable to or better than traditional token-based models*
> Additional context: *Large language models typically operate at the level of tokens (from some fixed vocabulary) rather than at the level of individual characters. However, compared to token-level models, character-level models have a number of advantages in their ability to perform tasks involving character-level information, such as recognizing small spelling mistakes or counting the number of occurrences of a character in a word. It would therefore be desirable to be able to move away from tokenization as a paradigm and towards character-level models; however, it is challenging to make character-level models as compute-efficient as token-level models, because modeling text at the character level results in much longer sequence lengths. I am interested in any recent papers (2019 or later) which demonstrate techniques for character-level language modeling with efficiency (either training efficiency or inference efficiency, or both) comparable to tokenization-based alternatives.*

> Topic: *Experiments that use tapered optical fibers to couple light into a microfabricated waveguide in the visible spectrum*
> Additional context: *Tapered optical fibers take the mode from the fiber core to largely being evanescent and can be used to couple into other waveguides with high efficiency. I am curious about how these tapered fibers are mechanically attached when this method is used. I care most about results which use light in the visible spectrum, so between 400 nm and 800 nm wavelength.*

These complex searches involve many concepts: For the latter search, relevant papers must contain experimental not theoretical results, use tapered optical fibers, talk about optical coupling into a microfabricated waveguide, and must use visible spectrum light. In addition, the user clarifies they

[6] Someone asking for "any quantum experiment" would find thousands of papers, with a very long time constant for exponential saturation, while someone asking for a very specific topic might find only 1 or 0 papers, with a short time constant.

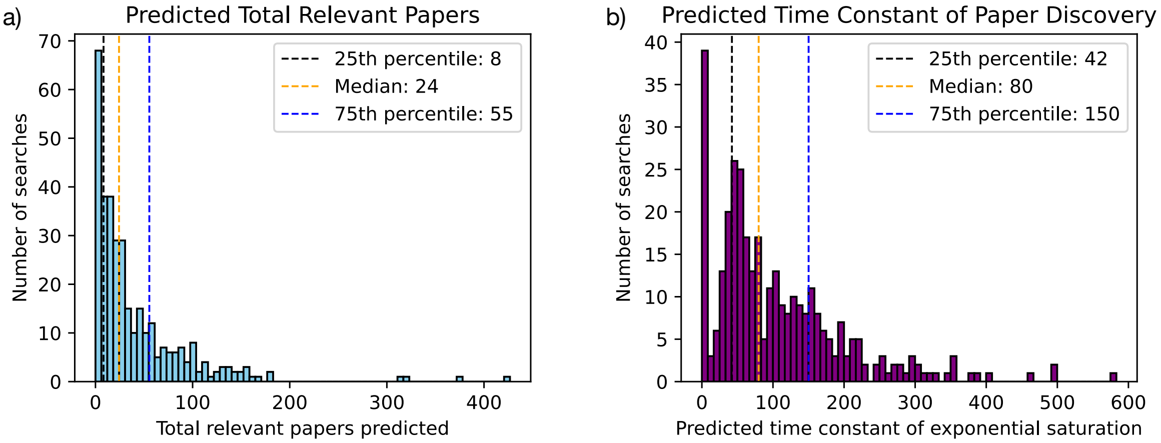

Figure 3: **Statistics of Undermind user searches.** Histograms of the exponential amplitude (a) and time constants (b) for the best fits to the discovery curves (as in Fig. 1(b)) of ~ 300 user searches. (a) The amplitude Undermind predicts for each search is the total number of papers Undermind expects to find if the search is extended to fully converge. (b) The time constant τ of the exponential describes how quickly this exponential discovery process approaches convergence. The discovered fraction of total papers f after evaluating n papers is modelled as $f = 1 - e^{-n/\tau}$. The majority ($\sim 63\%$) of relevant papers are discovered after τ papers are evaluated. Typical Undermind searches evaluate 150 papers.

are most interested in learning about mechanical attachment methods. This is a very difficult, if not impossible, goal to convey to a keyword search engine, though such goals can be achieved by Undermind.

3.2 Benchmarking Undermind's relevance classification accuracy

Table 1 presents measurements of the classification accuracy of the Undermind language model classifier, the second step of the search algorithm described in Sec. 1.1. This classifier looks at the user's request (topic, and any additional context they provide), and portions of the full text of an ArXiv paper, and decides whether that paper is "highly relevant", "closely related", or can be ignored. This classification accuracy was benchmarked by manually analyzing over 400 papers across a range of representative searches, and comparing the human evaluation to the language model's judgment.

We present the conditional probabilities of misclassification in Table 2. If a paper is highly relevant as judged by humans, there is a ∼2% chance it is identified as not relevant by the Undermind classifier. If a paper is closely related as judged by humans, there is a ∼9% chance it is identified as not relevant. Conversely, if Undermind says a paper is highly relevant, there is a less than 4% chance a human would say it is irrelevant (indicating a low probability of wasting time reading Undermind's results). The outcome: **a user can confidently only read the presented highly relevant and closely related results**, without fear of missing any highly relevant results, and without wasting time reading irrelevant papers.

The accuracy of Undermind's classifier was crucial for the analysis in this report: it makes it possible to automate the classification of more than 2,500 Google Scholar retrieved papers in parallel, which would otherwise take ∼ 50 hours of human effort.

3.3 Methodology for generating keyword phrases for Google Scholar comparison

When formulating their queries for Undermind, scientists were told to phrase their request to capture the entirety of their search goals and conditions. As a result, many of their queries are verbose and complex, and un-optimized for keyword search (see Appendix 3.1 for examples).

In order to translate these verbose queries into a format usable by Google Scholar for our comparison, we needed to mirror the process a human takes to break down their complex search task into

Undermind Classification	**Human Judgment**		
	Highly relevant	Closely related	Not relevant
Highly relevant	85	17	0
Closely related	25	72	8
Not relevant	2	9	214

Table 1: **Statistics of Undermind's classifications compared to human judgement.** We analyzed 432 papers classified by the language model, and carefully checked which classification a human rater would independently assign each paper, given the user's request, across a range of searches. Each cell shows the number of papers which the language model classified into a specific category (the row) and which the human classified into a specific category (the column). Note that, if we continued to analyze more irrelevant papers for a given search, the bottom right cell would increase indefinitely, while other cells would remain saturated at fixed values. This is because "not relevant" papers with very low ranking have virtually no false positive or false negative events, because the language model can clearly identify they are off topic.

Human Judgment	**Undermind Classification Probability**		
	Highly relevant	Closely related	Not relevant
Highly relevant	$75.9\%^{+9.8}_{-6.4}$	$22.0\%^{+9.6}_{-6.2}$	$1.8\%^{+4.7}_{-1.0}$
Closely related	$17.3\%^{+9.7}_{-5.8}$	$73.0\%^{+10.9}_{-7.0}$	$9.2\%^{+8.0}_{-4.1}$

Undermind Judgment	**Human Classification Probability**		
	Highly relevant	Closely related	Not relevant
Highly relevant	$83.3\%^{+9.4}_{-5.6}$	$16.7\%^{+9.4}_{-5.6}$	$0.0\%^{+3.8}_{-0.0}$
Closely related	$24.0\%^{+10.2}_{-6.5}$	$69.0\%^{+10.9}_{-7.2}$	$7.6\%^{+7.2}_{-3.6}$

Table 2: **Conditional classification rates.** Top: Undermind classification probabilities conditioned on human judgements. Associated upper and lower 95% confidence intervals are shown. Bottom: Human classification probabilities conditioned on Undermind judgement of a paper as highly relevant or closely related. For each table, note the far right column, which gives the probability that a truly relevant paper is missed (upper table) or the probability that a paper emphasized by Undermind is irrelevant (lower table).

bit-sized keyword searches. To automate this process, we prompted GPT-4 to create 5 keyword search phrases from each Undermind query (prompt details below). We then gathered the top 10 papers found by each of these keyword searches on Google Scholar (50 total papers) to compare to the papers Undermind retrieves and analyzes.

Generating keyword search phrases Here is an example of how GPT-4 was used to generate the keyword search phrases for a user search:

> Topic (user-provided): *Comprehensive overviews of the development of large language model architectures over time*
>
> Additional context (user-provided): *I am interested in finding papers that explicitly provide and overview of the major advances made in designing large language models (LLMs) (primarily the transformer architecture, but also others if applicable). I want to find papers that specifically discuss the research advances, and review the major papers published and models developed by academic and industrial labs and their contributions, for example (a paper needn't cover absolutely all of these, topics but this sort of thing should be the focus). I do NOT want papers that ONLY talk about what LLMs are, or review how they work. I am most interested in papers published in 2023 (please highlight these, but also include papers from other years.)*
>
> Prompt to GPT-4 to generate search phrases:

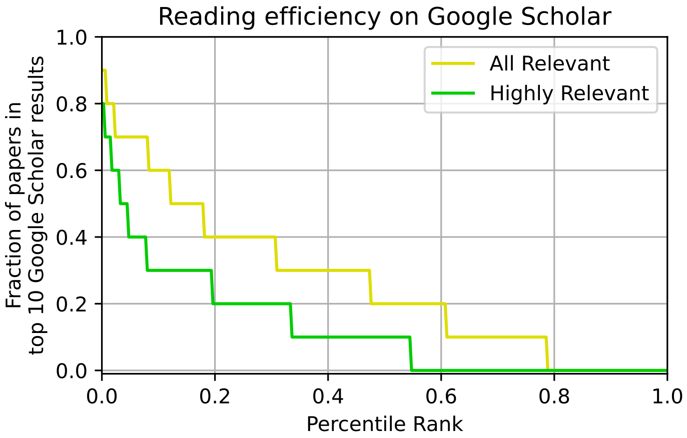

Figure 4: **Benchmarking the efficiency of reading Google Scholar's top results.** This plot shows the fraction of relevant papers found within the top 10 ArXiv papers returned by Google Scholar for each of the $\sim$ 300 user-generated searches (ordered by percentile on the x-axis). For most searches, the relevant results are sparse.

System message: *You are a thoughtful, expert scientist, and you are knowledgeable about carefully crafting a search phrase to find useful papers in a search engine.*

User message: *I am trying to help a colleague find papers about this topic: '{topic}'. In addition, here is some background information they provided: '{additional_context}'. I want you to help me generate 5 unique keyword searches for Google Scholar that will find papers on this topic. Do not use boolean operators. Make sure not to repeat searches without changing the keywords. Make some searches broad and some narrow, some very short, and some slightly longer.*

Generated keyword search phrases for this example:

1. *evolution of language model architectures*
2. *historical review transformer language models*
3. *large language models development milestones*
4. *language model architecture advancements 2023*
5. *comparative study large language model architectures*

For each of these keyword search phrases, we gathered the top 10 results from Google Scholar (top page). We then found the ArXiv papers in these results (typically $\sim$ 30 papers out of 50 gathered). We ordered these ArXiv papers in a round-robin fashion (top paper from one search, then the top from the next, and so on). We refer to the first 10 ArXiv papers discovered as the effective "first page" of Google Scholar, and when we quote the "top 5 pages of Google Scholar", we are referring to all ArXiv papers found in the entire 50 results.[7] We believe this set of ArXiv papers from the "top 5 pages" is a reasonable approximation of the set of papers a human could parse with significant manual effort.

3.4 Measuring sparsity of relevant works within Google Scholar's top results

We evaluated the first 10 ArXiv papers found by Google Scholar using Undermind's high quality classification system to determine if each paper was relevant to the user's original request. Fig. 4 shows the fraction of these top 10 results which were actually relevant to a user's search, across the full set of Undermind searches.

When reading through the top few Google Scholar results, often more than 90% of results are completely irrelevant (right side of graph). Note that relevant papers do exist for most of these

[7] In 4 out of the $\sim$ 300 searches, we found less than 10 ArXiv IDs in the top 5 pages of Google scholar (these searches only had 7, 8, 9, and 9 ArXiv papers). For simplicity, we treat these searches as if we had found 10 ArXiv papers to analyze and classified these few additional papers as irrelevant.

searches (see Fig. 2 for the predicted number of relevant papers for most searches). Google Scholar simply finds very few of these relevant papers.

3.5 Demonstrating exhaustiveness of Undermind's converged searches

3.5.1 General outline of how to measure exhaustiveness

Method 1: evaluating the entire database In order to demonstrate Undermind's exhaustiveness, one would ideally *evaluate the entire database* of papers by taking the following steps:

1. Gather many fully converged Undermind searches, where Undermind predicts it has found everything.[8]

2. For each of those same searches, check every single other paper in the database to see if it is relevant (all 2.3 million ArXiv papers).

3. Directly report the fraction of true relevant papers missed by Undermind for the average search.

Method 2: ensembling search methods Because evaluating every paper is prohibitively expensive, a different approach is usually taken. Instead, one *samples many complementary search methods* which are somewhat uncorrelated. Their combined results are assumed to exhaustively gather all relevant papers. One can then compare the retrieved papers of any specific search method to the set of all papers found by all the methods. The advantage of this approach is that one only needs to evaluate a small fraction of all papers in the database to find all truly relevant results.

Method 3: comparing two search methods A final method to evaluate exhaustiveness is to *compare the fraction of relevant results mutually found by two semi-independent methods.* An outline follows: Assume method A and method B of finding relevant papers are uncorrelated (by uncorrelated, we mean the two methods draw the relevant papers they return independently from a hypothetical larger set of all relevant papers), and that method A finds a fraction α of all total relevant results (this is the parameter we would like to estimate). If one examines the relevant results found by method B, a fraction α of these relevant results in B will be found by A as well due to randomness. One can thus estimate α, the fraction of all results that method A finds, as:

$$\alpha = \text{Exhaustiveness of } A \approx \frac{\text{Relevant results found simultaneously by method } B \text{ and method } A}{\text{All relevant results found by method } B} \tag{1}$$

This is the method we use to benchmark Undermind.

3.5.2 Estimating Undermind's exhaustiveness by comparing to Google Scholar

We use the method of comparing semi-independent search methods (see equation (1)), to estimate the exhaustiveness of Undermind. We compare exhaustive Undermind searches to the papers retrieved in the top 10 results from Google Scholar for that same search. Papers "encountered" by Undermind refer to papers that Undermind decided to check with its relevance classifier. The data required to calculate exhaustiveness are derived from Fig. 5, and are summarized here:

- Within the top 10 results from Google Scholar, for a converged Undermind search, Google Scholar typically finds 5.25 papers that were never encountered by Undermind (Fig. 5(a), right side).

- Within the 5.25 papers unencountered by Undermind, only 0.03 highly relevant papers are found on average (Fig. 5(c), right side).

- Within the 4.75 papers that were already encountered by Undermind (Fig. 5(d), right side), 1.21 are highly relevant (Fig. 5(f), right side).

[8] These are searches where Undermind predicts it will not discover more relevant papers with further reading, because it has read enough papers to saturate the exponential discovery curve.

To estimate the exhaustiveness of Undermind using equation (1), we take the ratio:

$$\frac{\text{Highly relevant papers found by Undermind in Google Scholar top 10}}{\text{All highly relevant papers in Google Scholar top 10}} = \frac{1.21}{1.21 + 0.03} \approx 97.6\%. \quad (2)$$

It is therefore likely that *a converged Undermind search contains essentially all highly relevant papers that exist*, within a few percent statistical error.[9]

One can also estimate the exhaustiveness of finding closely related papers with Undermind, though this is a less crucial metric (since these papers are not precisely about the topic). Using the above methods, Undermind appears to find $1.13/(1.13 + 0.28) \approx 80\%$ of all closely related papers that exist. However, this is likely a significant underestimate because of the misclassification rate of Undermind's relevance classifier. These misclassification errors are difficult to estimate, and contain significant uncertainty, so we omit a rigorous analysis of the true exhaustiveness of closely related papers.[10]

3.6 Measuring the total number of relevant papers in the top 50 Google Scholar results

To save on compute costs, instead of running the relevance classifier over all the ArXiv papers found in the top 50 papers on many Google Scholar searches, we can obtain a close estimate of the number of relevant hits in the top 50 Google Scholar results using the data in Fig. 5.

For a converged Undermind search, we established in Appendix 3.5 that Google Scholar finds virtually no relevant papers Undermind misses. Therefore, one can use the set of Undermind-discovered relevant papers as the ground truth, and simply check how many of those same papers appear in the top 50 results of Google Scholar.

For non-converged searches, we can still easily estimate the number of expected relevant papers in Google Scholar's top 50 results using the data in Fig. 5. To do so, we first estimate the fraction of total papers a given search has found so far, which places the search at a given position on the x-axis of Fig. 5. At that x position, we next estimate the ratio

$$\frac{\text{Total relevant papers in Google Scholar top 10}}{\text{Relevant papers found by Undermind in Google Scholar top 10}} \quad (3)$$

by comparing the best fit data in Fig. 5(b-c) to Fig. 5(e-f). Finally, we count the number of relevant papers that the non-converged Undermind search has found in the top 50 results, and correct this upwards to account for the undiscovered papers. This correction factor is in the range of 1× to 2.5×. Where necessary, the data shown in Fig. 2 have this correction already applied.

[9] Sources of uncertainty include: error on the best fit lines in Fig. 5, misclassification errors of Undermind in Table 1 and Table 2, and the uncertainty from a finite sample size of ~ 300 highly relevant papers.

[10] As an outline, misclassification of irrelevant papers as closely related occurs at a $\sim 4\%$ rate in Table 1. Assuming this 4% error also holds for Google Scholar sampled papers (not necessarily justified), this implies the ~ 5 irrelevant papers in the set of unencountered papers would produce ~ 0.2 falsely identified closely related papers, a large fraction of the observed 0.28 closely related papers in Fig. 5(b), right side.

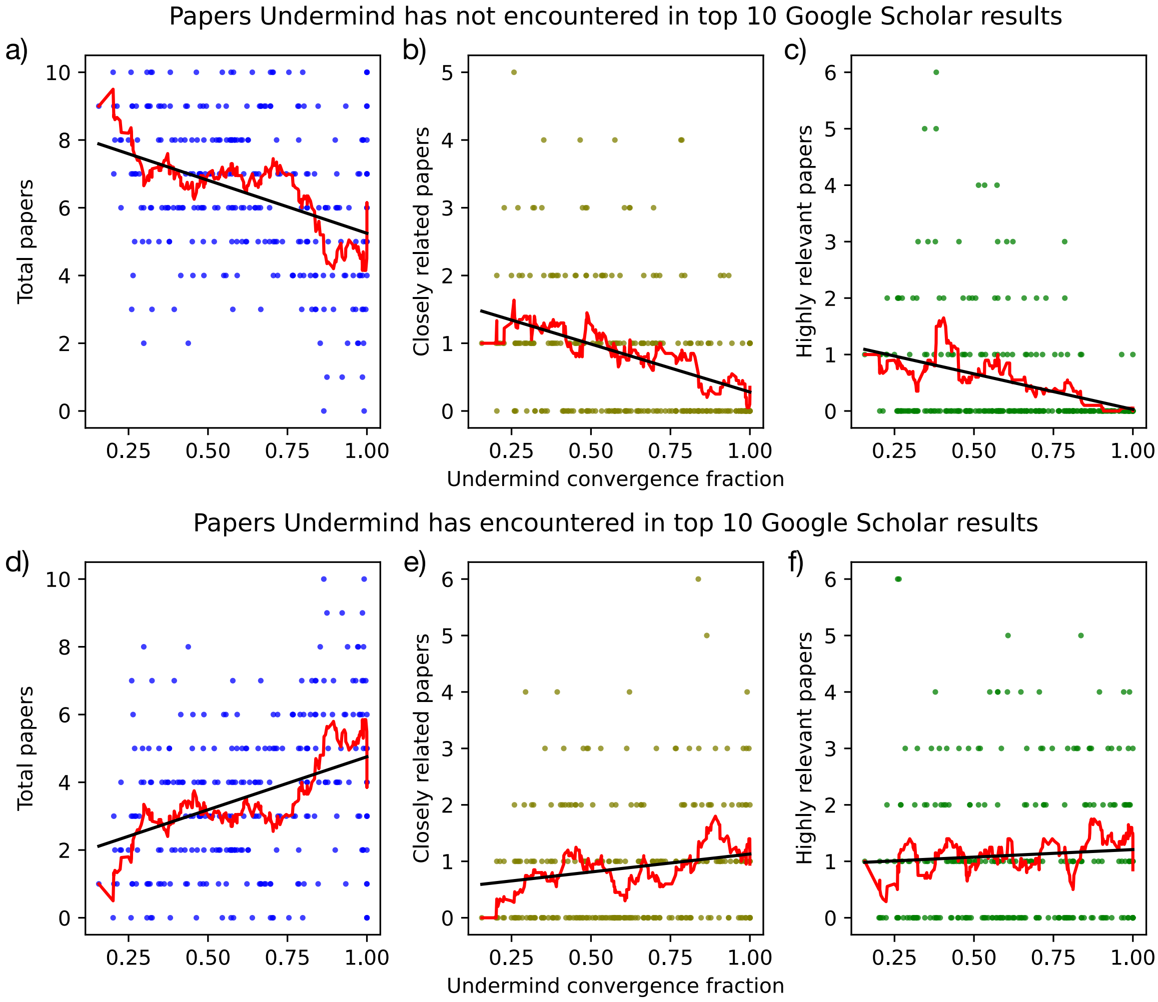

Figure 5: **Statistics of papers in the top 10 Google Scholar results.** These plots show the number of papers in the top 10 papers returned by Google Scholar which were not yet encountered and classified by Undermind (a), and how many of those were closely related (b) or highly relevant (c) after evaluating them with the language model classifier. These are shown as a function of the convergence fraction $f = 1 - e^{-n/\tau}$ of each Undermind search, which is Undermind's best estimate of the fraction of relevant papers it has found so far (f is described further in Fig. 3). Red lines show moving averages of 20 datapoints, and black lines are best fit lines to the entire dataset. (d-f) shows the same corresponding data for the papers that were already encountered by Undermind in the top 10 Google Scholar results. (a-c) show that converged searches (far right of each graph) have on average ~ 5 papers in the top 10 which Undermind has not yet encountered and evaluated. However, virtually no new highly relevant papers are discovered when those papers are evaluated. See Appendix 3.5 for further details and interpretation.