SciMON 🧪: Scientific Inspiration Machines Optimized for Novelty

Qingyun Wang[1], **Doug Downey**[2], **Heng Ji**[1], **Tom Hope**[2,3]
[1] University of Illinois at Urbana-Champaign [2] Allen Institute for Artificial Intelligence (AI2)
[3] The Hebrew University of Jerusalem
{tomh,doug}@allenai.org, {qingyun4,hengji}@illinois.edu

Abstract

We explore and enhance the ability of neural language models to generate novel scientific directions grounded in literature. Work on literature-based hypothesis generation has traditionally focused on binary link prediction—severely limiting the expressivity of hypotheses. This line of work also does not focus on optimizing novelty. We take a dramatic departure with a novel setting in which models use as input background contexts (e.g., problems, experimental settings, goals), and output *natural language ideas* grounded in literature. We present SciMON, a modeling framework that uses retrieval of "inspirations" from past scientific papers, and explicitly optimizes for novelty by iteratively comparing to prior papers and updating idea suggestions until sufficient novelty is achieved. Comprehensive evaluations reveal that GPT-4 tends to generate ideas with overall low technical depth and novelty, while our methods partially mitigate this issue. Our work represents a first step toward evaluating and developing language models that generate new ideas derived from the scientific literature[1].

1 Introduction

Can machines mine scientific papers and learn to suggest new directions? The idea that information from the literature can be used for automatically generating hypotheses has been around for decades (Swanson, 1986). To date, the focus has been on a specific setting: hypothesizing links between pairs of concepts (often in drug discovery applications (Henry and McInnes, 2017), e.g., new drug-disease links), where concepts are obtained from papers or knowledge bases previously derived from papers (Sybrandt et al., 2020; Nadkarni et al., 2021).

This common setting has fundamental drawbacks. Reducing the "language of scientific ideas" (Hope et al., 2023) to this simplistic form limits the expressivity of the hypotheses we can hope to generate, and does not capture nuanced *contexts* that scientists consider: target application settings, requirements and constraints, motivations and challenges. In light of the strong progress recently made with large language models (LLMs), in this paper we explore a dramatically different setting: models that take descriptions of problem contexts—and return *natural language* suggestions of novel scientific directions that are grounded in literature.

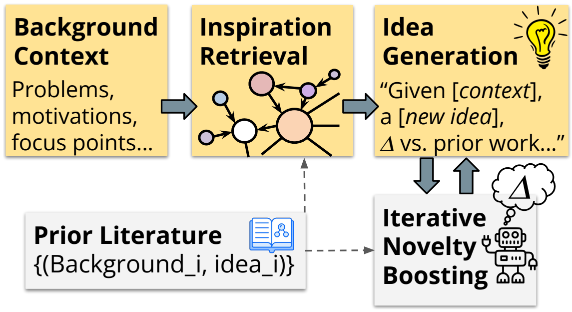

Figure 1: SciMON takes background context and generates ideas grounded in literature inspirations, optimizing novelty by iteratively comparing to related work.

We develop a framework named SciMON (Scientific Inspiration Machines with Optimization for Novelty), named after Nobel laureate and AI pioneer Herbert Simon who authored early foundational work on automated scientific discovery (Newell and Simon, 1956; Simon, 1973). We first present an automated data collection methodology that collects examples of past problems and proposed ideas from scientific papers. We then use this data for both fine-tuning and in-context training of LLMs—training them to take problem descriptions and output proposed ideas to address them. We observe that state-of-art LLMs (e.g., GPT-4 (OpenAI, 2023)) struggle with generating novel scientific ideas, and contribute a new modeling framework for generating hypotheses that makes progress in improving the hypothesis generation

[1]Code, data, and resources are publicly available for research purposes: https://github.com/eaglew/clbd.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 279–299
August 11-16, 2024 ©2024 Association for Computational Linguistics

ability of LLMs (Figure 1). Given a background problem description, models first dynamically retrieve *inspirations* from past literature in the form of related problems and their solutions along with contexts from a scientific knowledge graph. These retrieved inspirations serve to ground the generated ideas in existing literature. We then endow models with the ability to iteratively boost the *novelty* of generated ideas. Given an idea $\mathcal{I}$ generated by the LLM at step t, the model compares $\mathcal{I}$ with existing research in the literature; if it finds strongly overlapping research, the model is tasked with updating its idea to be more novel relative to prior work (much like a good researcher would do). We also introduce an *in-context contrastive model* which encourages novelty with respect to background context.

We perform the first comprehensive evaluation of language models for generating scientific ideas in our new generative, contextual setting. We focus on AI/NLP ideas to facilitate analysis by AI researchers themselves, and also demonstrate generalization to the biomedical domain. We design extensive evaluation experiments using human annotators with domain expertise to assess relevance, utility, novelty, and technical depth. Our methods substantially improve the ability of LLMs in our task; however, analyses show that ideas still fall far behind scientific papers in terms of novelty, depth and utility—raising fundamental challenges toward building models that generate scientific ideas.

2 Background and New Setting

We begin with a brief description of related work and background. We then present our novel setting.

Literature-based discovery Nearly four decades have passed since Don Swanson pioneered Literature-Based Discovery (LBD), based on the premise that the literature can be used for generating hypotheses (Swanson, 1986). LBD has been focused on a very specific, narrow type of hypothesis: links between pairs of concepts (often drugs/diseases). The classic formalization of LBD goes back to Swanson (1986) who proposed the "ABC" model where two concepts (terms) A and C are hypothesized as linked if they both co-occur with some intermediate concept B in papers. More recent work has used word vectors (Tshitoyan et al., 2019) or link prediction models (Wang et al., 2019; Sybrandt et al., 2020; Xu et al., 2023) to discover scientific hypotheses as pairwise links between concepts. A tightly related body of research focuses on *scientific knowledge graph link prediction* (Nadkarni et al., 2021), where predicted links may correspond to new hypotheses, and knowledge bases are reflections of existing scientific knowledge in specific domains, derived from literature. A fundamental gap in this line of work is in the lack of approaches for modeling nuanced contexts (Sosa and Altman, 2022) (e.g., the specific settings in which a drug may be relevant for a disease) for generating ideas in open-ended problem settings with unbounded hypothesis spaces, and for optimizing novelty. Our setting can be viewed as a radical departure addressing the limitations in existing settings.

LLMs for Scientific Innovation Large language models (LLMs) have made remarkable progress in interpreting and producing natural language content and handling knowledge-intensive tasks such as in the medical domain (Nori et al., 2023). Very recent work (Boiko et al., 2023) has explored the use of LLMs in a robotic chemistry lab setting, planning chemical syntheses of known compounds and executing experiments. Robotic lab settings are inherently limited to narrow sub-areas where such experiments are possible and relevant. Other very recent work (Huang et al., 2023) used LLMs to produce code for machine learning tasks such as Kaggle competitions, finding that a GPT-4 agent achieved 0% accuracy on research challenges such as BabyLM (Warstadt et al., 2023). GPT-4 has been anecdotally reported as having "strengths less like those of having a human co-author, and more like a mathematician working with a calculator" (Carlini, 2023). Our goal is to conduct a non-anecdotal evaluation and enhancement of strong LLMs' ability to generate novel open-ended scientific ideas.

2.1 SciMON Problem Setting

We are motivated by imagining an AI-based assistant that suggests ideas in natural language. The assistant takes as input background context $\mathcal{B}$ consisting of (1) current problems, motivations, experimental settings and constraints, denoted as $\mathcal{M}$; and optionally (2) a seed term v that should be a focus point of the generated idea $\mathcal{I}$. The seed term is motivated by considering a user-provided cue for the model to limit its hypothesis space. Importantly, generated ideas should not merely paraphrase the background—the output should be *novel* with respect to $\mathcal{B}$ and the broader literature corpus. Figure 2 illustrates the setting, showing a background

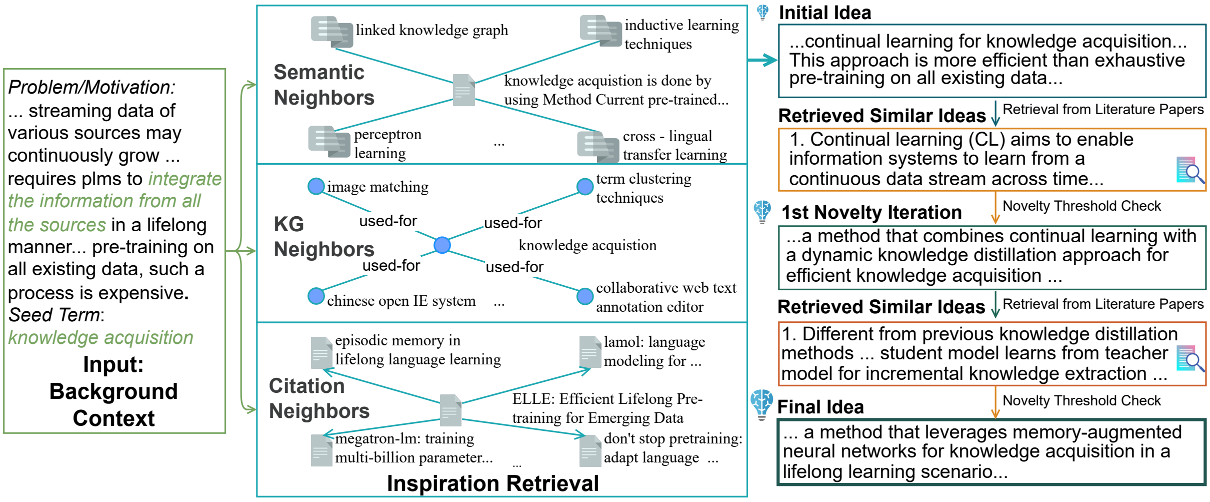

Figure 2: Architecture overview. Our models retrieve *inspirations* and then pass the background input and retrieved inspirations to an LM-based generation module, which iteratively optimizes novelty. Input from Qin et al. (2022).

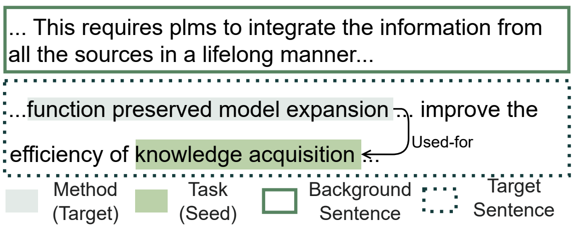

Figure 3: We use IE to obtain literature data for our approach: problems/motivations (background) and proposed ideas (target), as well as salient seed terms.

text that describes problems with *"pretrained language models"* in the lifelong integration of information sources, including computational costs. The assistant aims to generate an idea for performing *"knowledge acquisition"* within this context. Given this input, we aim to generate a full sentence describing a novel idea.

2.2 Automated Training Data Collection

We obtain training data derived from papers with scientific information extraction (IE) models—extracting past examples of background sentences and corresponding ideas (e.g., descriptions of methods used for specific problems in the background sentences), along with salient entities as seed terms. This data is used for training in both in-context learning and fine-tuning setups.

We construct a corpus $\mathcal{D}$ from 67,408 ACL Anthology papers from S2ORC (Lo et al., 2020) (we later also conduct an experiment with a biomedical corpus §4.1). Given a title and the corresponding abstract from a document d, to select problem/motivation sentences $\mathcal{M}$ we first perform scientific sentence classification (Cohan et al., 2019) to classify sentences from the abstract into one of {*Background*, *Method*, *Objective*}, selecting sentences with labels of *Background* and treating the remaining sentences as target sentences $\mathcal{T}$ which will serve as desired output examples (Figure 3).

For seed term selection, we apply a state-of-the-art scientific IE system (Ye et al., 2022) to $\mathcal{T}$ to extract *entities* corresponding to *Task*, *Method*, *Evaluation Metric*, *Material*, and *relations* of the form `[method,used-for,task]`—mentions of methods and the tasks they are used for, materials used for tasks, etc. We treat the head (e.g., method) or tail (e.g., task) entity as the *seed* term, and name the other entity (tail/head, respectively) as a *target* term $t \in \mathcal{T}$. Continuing our example from Figure 2, Figure 3 shows how the seed and target terms ("*knowledge acquisition*" and "*function preserved model expansion*") are extracted from $\mathcal{T}$. During training, each instance contains ($\mathcal{B}$,$\mathcal{T}$) pairs; during evaluation, target information is removed.

We use SciCo (Cattan et al., 2021) to obtain coreference links for entity normalization, and use ScispaCy (Neumann et al., 2019) to replace abbreviations with a more informative long form. We also collect paper metadata, including the citation network $\mathcal{G}_c$. We split our dataset temporally (train/dev/test correspond to papers from years <2021 / 2021 / 2022 respectively). For our experiments, we used model checkpoints trained on data preceding 2022, avoiding the risk of data contamination (§6). Table 1 shows data statistics.[2]

[2]More details are in Appendix C.

Quality of IE Preprocessing During preprocessing, we only keep high-confidence outputs from IE models to reduce errors. We observe this removes many of the noisy cases. To validate this, we manually evaluate the precision of each preprocessing step on a random sample of papers and observe that all steps yield high precision (91%-100%) except relation extraction (65%); in total, the rate of instances passing all steps was 79.7%.[3]

Gold Test Set We create a high-quality, clean test set. We remove test instances where models can trivially use surface-level background information to infer the ground truth to create a more *challenging set*, selecting instances with low similarity between background and ground truth sentences. We compute the cosine similarity between each instance's background and corresponding ground truth sentence in the test set and take pairs with similarity ≤ 0.074, which amounts to the tenth percentile of pairs. We further annotate this subset to create a *gold subset*. We manually exclude instances with trivial overlap between ground truth and background, remove cases with irrelevant background, and retain only instances where the target relation (from which the seed term is taken) is salient to the target sentence. We also remove test pairs that have unexplained terms in the background. We obtain a total of 194 instances.[4]

Split	Forward	Backward	Total
Train	55,884	58,426	114,310
Valid	7,938	8,257	16,195
Test	2,623	2,686	5,309

Table 1: Dataset statistics. Considering a relation of the form [v `used-for` u], we define [v `used-for ?`] as *forward*, and [`? used-for` u] as *backward*.

3 SciMON Models

We present a new module to retrieve inspirations as contextual input (§3.1). Then, we describe another module to generate ideas given the context+inspiration (§3.2). Finally, we introduce a new iterative novelty optimization method to further improve idea quality (§3.3).[5]

[3]See Table 6 in Appendix.
[4]Full annotation details are in Appendix C.
[5]Training and hyperparameter details in Appendix B.

3.1 Inspiration Retrieval Module

We take broad inspiration from cognitive aspects of innovation (Hope et al., 2023): when researchers generate a new idea, they are grounded in a web of existing concepts and papers bearing on the new idea. We aim to enrich the context of each background by retrieving "inspirations"— pieces of information that can guide hypothesis generation. As illustrated in Figure 2, for a given instance of the SciMON task, our retrieval augmentation can retrieve from three types of sources. Each source uses a different form of query and output.

Semantic Neighbors For a given problem/motivation as input, ideas proposed for related problems in the training set can serve as a guiding reference for generating a new idea. Given the background context $\mathcal{B}$ with a seed term v and problem/motivation $\mathcal{M}$, we construct a base input b: a concatenation of $\mathcal{M}$ with a prompt $\mathcal{P}$ belonging to one of two templates: "v is used for p" or "v is done by using p", where p is one of *Task/Method/Material/Metric*. In short, $b := \mathcal{P} \oplus \textit{context:}\mathcal{M}$. For example, in Figure 2, the concatenation is "*Knowledge acquisition is done by using Method; Context:...requires plms to integrate information...lifelong manner...*".

We then retrieve inputs from the training set that are semantically related to a new base input b, and obtain target sentences T corresponding to each retrieved training input. We extract the target term $t \in \mathcal{T}$ matching the seed term in b (§2.2) as inspiration for input b. Simply put, this means we use as inspiration the salient aspect of the solution proposed in $\mathcal{T}$, which we found empirically to help remove noisy/irrelevant information in $\mathcal{T}$. For example, in Figure 2, we find *"informative entities are done by using Method context: in this work, we aim at equipping pre-trained language models with structured knowledge."* as similar to the input and use $t =$*"linked knowledge graph"* as inspiration.

Technically, we first construct a fully connected graph $\mathcal{G}_S$ based on the training set where each node is a pair of input text b_i and target term t_i. We define the weight between two nodes i and j as the cosine similarity between b_i and b_j based on representations from SentenceBERT (Reimers and Gurevych, 2019) (`all-mpnet-base-v2`). Given b, we first insert it into $\mathcal{G}_S$ and compute the weights of its connected edges. We then retrieve neighbors input text $\{b_1, \ldots, b_k\}$ from the training set with the largest edge weight, where k is the number of retrieved instances. We consider the corresponding target terms $\{t_1, \ldots, t_k\}$ as semantic inspirations.

KG Neighbors We also explore enriching the context by linking it to a background KG with information on related methods and tasks. Using the same IE process used to extract our training examples (§2.2), we create a *global* background KG $\mathcal{G}_B$ which covers all papers in the corpus $\mathcal{D}_\mathcal{Y}$ prior to a given year $\mathcal{Y}$ (i.e., the nodes in $\mathcal{G}_B$ correspond to tasks/methods/materials/metrics, and the edges are used-for relations, extracted and normalized from across the entire corpus as described earlier). Then, given a seed term v at query time, we select adjacent nodes $\{n_1, n_2, ...\}$ from $\mathcal{G}_B$ as inspirations. As an example, in Figure 2, the neighbor nodes of *"knowledge acquisition"* include *"collaborative web text annotation editor"*, *"image matching"*, etc., which we select as inspirations.

Citation Neighbors Another notion of contextual relatedness we explore is via citation graph links. Here, given as input background context $\mathcal{B}$, we assume access to the original source document d from which $\mathcal{B}$ was extracted, and consider its *cited* paper title set $\mathcal{C}_d$ as potential candidates. This can be seen as a stronger assumption on information available to the model— assuming a researcher using the model provides relevant candidate documents from which ideas could be pooled. Because the training set only contains papers before year $\mathcal{Y}$, we only select papers $\mathcal{C}_{d\mathcal{Y}} \subseteq \mathcal{C}_d$ prior to year $\mathcal{Y}$. We then retrieve the top-k titles with the highest cosine similarity to d from $\mathcal{C}_{d\mathcal{Y}}$ based on their SentenceBERT embeddings as earlier. For instance, in Figure 2, the paper ELLE (Qin et al., 2022) cites the paper (de Masson d'Autume et al., 2019). Therefore, we choose the title *"episodic memory in lifelong language learning"* as inspiration information.

3.2 Generation Module

The idea generation module is given retrieved inspirations $i_1, \ldots, i_k$ along with context $\mathcal{M}$ as input.

In-Context Learning We experiment with recent state-of-the-art LLMs, GPT3.5 `davinci-003` (Ouyang et al., 2022) and GPT4 `gpt-4-0314` checkpoint (OpenAI, 2023). We first ask the model to generate sentences based on the seed term and the context in the zero-shot setting without any in-context examples (`GPT3.5ZS`, `GPT4ZS`). We then ask the model to generate sentences in a few-shot setting by prompting randomly chosen pairs of input and output from the training set (`GPT3.5FS`, `GPT4FS`). Inspired by Liu et al. (2022), we further employ a few-shot setting using semantically similar examples. Instead of random in-context examples, we use the top-k examples from the training set with the highest cosine similarity to the query (`GPT3.5Retr`). This few-shot retrieval setting differs from the semantic neighbor discussed above, in that we provide both the input and output of each instance rather than solely supplying target entities as additional input.

Fine Tuning We fine-tune T5 (Raffel et al., 2020) (more recent models may be used too; see our biomedical experiment §4.1 fine-tuning an LLM). We observe that the generation models tend to copy phrases from the background context. For example, given the context *"...hierarchical tables challenge numerical reasoning ..."*, the model will generate *"hierarchical table reasoning for question answering"* as the top prediction. For generating suggestions of *novel* ideas, we wish to discourage overly copying from the background context. We introduce a new in-context contrastive objective, where negative examples are taken from the text in the input (e.g., in Figure 2, the in-context negatives are *plms*, *pre-training*, etc). We compute an InfoNCE loss (Oord et al., 2018) over the hidden states of the decoder, aiming to maximize the probability of the ground truth against those of in-context negatives:

$$\begin{aligned} y^+ &= \sigma(\mathrm{Avg}(\mathbf{W}_y\mathbf{h}^+ + \mathbf{b}_y)) \\ y_k^- &= \sigma(\mathrm{Avg}(\mathbf{W}_y\mathbf{h}_k^- + \mathbf{b}_y)) \\ \mathcal{L}_{\mathrm{cl}} &= \frac{\exp\left(y^+/\tau\right)}{\sum_k \exp\left(y_k^-/\tau\right) + \exp\left(y^+/\tau\right)} \end{aligned} \tag{1}$$

where $\mathbf{h}^+$ and $\mathbf{h}_k^-$ are decoder hidden states from the positive and k-th negative samples, $\mathbf{W}_y$ and $\mathbf{b}_y$ are learnable parameters, σ is a sigmoid function, τ is a temperature hyperparameter, and $\mathrm{Avg}(*)$ denotes the average pooling function based on the target sequence length. We optimize with both contrastive loss $\mathcal{L}_{\mathrm{cl}}$ and the cross-entropy loss.

3.3 Iterative Novelty Boosting with Retrieval

We further improve the novelty of generated ideas with a new iterative *retrieve-compare-update* scheme. Conceptually, we consider a novelty-inducing penalty $\gamma_{\mathrm{nov}}(\mathcal{I}, \mathcal{R})$ that penalizes ideas $\mathcal{I}$ that are too "close" to existing ideas in literature reference examples $\mathcal{R}$. $\gamma_{\mathrm{nov}}(\mathcal{I}, \mathcal{R})$ is included during in-context learning and inference, providing numerical feedback in the form of a score reflecting similarity to existing work. We wish to minimize this score while ensuring $\mathcal{I}$ remains relevant to the background context $\mathcal{B}$; we do so iteratively by (1) retrieving related work from $\mathcal{R}$, (2) measuring degree of novelty, (3) instructing the model to update $\mathcal{I}$ to be more novel w.r.t $\mathcal{R}$, conditioning on $\mathcal{B}$.

Specifically, in our implementation, we construct a reference corpus $\mathcal{R}$ based on all papers in the training set. We then propose an iterative algorithm that compares generated ideas against $\mathcal{R}$. We start with the initial idea $\mathcal{I}_0$ generated by the generation module. At each time step t, we use the generated idea $\mathcal{I}_t$ as a query to retrieve k nearest ideas from the literature reference corpus $\mathcal{R} = \{R_1, ..., R_k\}$ based on SentenceBERT, with the top-k highest cosine similarity scores to $\mathcal{I}_t$ (we use $k = 20$). For each retrieved ground truth literature idea R_i, we compare its cosine similarity score S_i against a threshold μ (we use 0.6). We provide all the retrieved ground truth ideas $\hat{\mathcal{R}}$ that pass the threshold as additional negative examples for the large language models with the following instruction prompt: "*Your idea has similarities with existing research as demonstrated by these j sentences: $\hat{\mathcal{R}}$ Make sure the idea you suggest is significantly different from the existing research mentioned in the above sentences. Let's give it another try.*" We stop the iteration once all S_i are lower than μ. Figure 2 and Table 5 demonstrate novelty iterations.

4 Experiments

4.1 Human Evaluation

We present four human evaluation studies, exploring different facets of our problem and approach.

4.1.1 Study I: Comparing Outputs across Model Variants

We recruit six volunteer NLP experts with graduate-level education to rate the system. Raters are told to envision an AI assistant that suggests new paper ideas. We randomly select 50 instances (background+seed) from the gold subset. Each annotator receives ten instances, each paired with system outputs from different model variants (Table 2). We ask raters to assess idea quality by considering each output's relevance to the context, novelty, clarity, and whether the idea is reasonable (positive ratings are dubbed "helpful" as shorthand, indicating they pass the multiple considerations). We observe moderately high rater agreement.[6] Raters are blind to the condition, and system outputs are randomly shuffled across instances.

We instruct annotators to only provide positive ratings to ideas sufficiently different from the input context. In Study I, we ask raters not to anticipate groundbreaking novelty from the system but rather a narrower expectation of quality and utility; in Study II below, we enrich the analysis to examine *ranking* between top models and also "raise the bar" and compare to actual ideas from papers.[7]

In a preliminary experiment, we also collected human ratings for `GPT4-ZS` (zero-shot) vs. `GPT4-FS` (few-shot) using the same criteria, finding `GPT4-FS` ranked higher in 65% of cases, with the rest mostly tied; thus, zero-shot GPT-4 was left out of the remainder of study I and subsequent studies to reduce annotation effort and cost.

Results Overall, `GPT4FS` and `GPT4FS+KG` outperform other models by a wide margin (Table 2). Apart from GPT4, `T5+SN+CL` performs best compared to other baselines, given its stronger prior knowledge of useful similar background hypotheses. In general, GPT3.5 models performed worse than fine-tuned T5 and its variants, which echoes results in other work in the scientific NLP domain (Jimenez Gutierrez et al., 2022). GPT4 outputs tended to be longer, which may partially explain higher human preference.

Type	`3FS`	`3Rt`	`3FS+CT`	`3FS+KG`	`4`	`4+KG`	`T5`	`T5+SN`
H	33	25	16	33	73	66	22	48
U	67	75	84	67	27	34	78	52

Table 2: Percent (%) of total votes each system output receives from human raters. *H* denotes a helpful output, while *U* denotes an unhelpful output. "`3FS`" refers to the `GPT3.5FS`. "`3Rt`" refers to the `GPT3.5Retr`. "`4`" refers to `GPT4FS`, and "`4+KG`" refers to the `GPT4FS+KG`. "`T5+SN`" refers to the `T5+SN+CL`. `GPT4FS` and `GPT4FS+KG` are rated much higher. While `GPT4FS` has a slightly higher rating than the KG variant, a further human study reveals that `GPT4FS+KG` often leads to more technical depth (§4.1).

4.1.2 Study II: Comparing GPT4 Variants against Real Papers

We conduct a follow-up human study of close competitors `GPT4FS` and `GPT4FS+KG` with a subset of

[6]The agreement scores are in Table 13 Appendix C.

[7]Full evaluator guidelines are in Appendix C. The sample annotations are in Table 11.

the annotators to evaluate the incrementality and novelty of the generated ideas. In this study, model outputs are now *ranked*, unlike the binary classification of helpful/not in Study I. Suggestions are ranked according to the level of technical detail and innovation in comparison to each other—i.e., ranking which of GPT4FS and GPT4FS+KG had a higher degree of technical detail and novelty, or whether they are roughly the same (tied). Finally, outputs are rated versus the ground truth idea, according to whether or not the suggestions were roughly at the same level of technical detail and innovation as the original paper's idea, or *significantly lower*.

Results Overall, GPT4FS+KG is found to have higher technical detail in 48% of the compared pairs, and found to be less incremental (more novel) in 45% of the pairs. Among the remaining 52%/55% (respectively), the vast majority are ties, indicating that whenever GPT4FS+KG is not favored, it is of roughly the same quality as GPT4FS, but not vice versa. However, the most crucial aspect is comparing the results against the original ground truth idea on the quality of innovation. Here, we find that in 85% of comparisons, the ground truth is considered to have *significantly higher* technical level and novelty; and in the remaining 15%, the ground truth was ambiguous or lacking additional context from the paper abstract. This points to a major challenge in obtaining high-quality idea generations using existing state-of-the-art models.

4.1.3 Study III: Evaluation on Iterative Novelty Boosting

We conduct a fine-grained evaluation of our novelty mechanism with qualitative and quantitative evaluation of novelty. Specifically, we ask five annotators to further compare the novelty-enhanced results against the initially generated ideas. We randomly select 70 instances (background+seed) from the sentence generation gold subset. We ask annotators to check whether the new ideas are *different* than the initial ideas (e.g., adding new information or approaches), and whether they are more *novel* (i.e., a new idea can be different, but not necessarily more novel). Since GPT4FS+SN outperforms other models, for this model, we further instruct annotators to compare the novelty of the second iteration results against the first iteration results.

Results For SN, in the first iteration 88.9% of updated ideas are substantially different from initial ideas, and for 55.6% we are able to increase novelty/creativity (meaning that, e.g., if 100 examples were updated, we would gain 56 examples that are more novel). The 2nd iteration, further increases novelty for 57.8% of the ideas that continued to another iteration. For ideas not considered more novel after applying our method, we do not observe a drop in novelty—the method either increases or maintains novelty.

Type	GPT4FS	+SN	+CT	+KG
1st Novelty Δ (%)	+54.4	+55.6	+47.8	+46.7
2nd Novelty Δ(%)	-	+57.8	-	-
1st new terms Δ	+23.1	+22.8	+22.1	+21.9
2nd new terms Δ	-	+21.5	-	-

Table 3: Relative improvements of iterative novelty boosting. Iterations are applied to the ideas for which sufficiently similar related work is detected (§3.3). "1st Novelty" is % of the 1st iteration ideas that gained *novelty* over the initial idea, and "2nd Novelty" is the % of gain over the 1st iteration. Our method substantially increases novelty for ideas to which it is applied. To save annotation resources, we only annotate second iteration results for the best-performing method (SN). We report the average number of new terms added, after filtering.

Ideas after novelty iterations are longer than initial ideas. We examine the new terms added after filtering 359 words, including stopwords, as many generic words and terms are often added (e.g., "novel model/method/approach"). While our method helps boost novelty, overall the model often tends to suggest combinations between popular concepts (§4.2). Novelty boosting seemed to often focus on adding dynamic/adaptive modeling, graph models and representations, the fusion of multiple modalities and sources—and sometimes all at once (e.g., "*Dynamic Syntax-Aware Graph Fusion Networks (DSAGFN)*"), and to explicitly compare against existing ideas from literature (Table 5).

Type	Meditron	+SN	+CT	+KG
Helpful(%)	35	80	60	50
Unhelpful(%)	65	20	40	50
vs. GT(%)	30	45	50	35

Table 4: Human evaluations results of each system output for the idea sentence prediction task on Biomedical Domain. "vs. GT" refers to percents which system outputs are better than ground truth ideas.

4.1.4 Domain Generalization Case Study

Our domain-agnostic framework can be applied to other domains by changing the IE system used in the preprocessing procedure. To demonstrate

Type	Content
Input (Dong et al., 2022)	*seed term*: **speech unit boundaries** ; *context* (abridged): ... generate partial sentence translation given a streaming speech input. existing approaches ... break the acoustic units in speech, as boundaries between acoustic units in speech are not even. ...
Initial idea	A pause prediction model to identify **speech unit boundaries** ...
Iteration 1	A method that leverages acoustic and linguistic features to predict **speech unit boundaries** dynamically, ensuring smooth transitions ... differs from the existing research as it combines both acoustic properties and linguistic context ... adapting to variations in speaker characteristics, speaking styles, and languages.
Iteration 2	A novel method called Adaptive **Speech Unit Boundary** Detection (ASUBD) ... a combination of attention mechanisms to focus on relevant acoustic and linguistic features and reinforcement learning to guide the system to make optimal predictions of unit boundaries based on previous decisions...
Ground Truth	... an efficient monotonic segmentation module ... accumulate acoustic information incrementally and detect proper **speech unit boundaries**.

Table 5: Example of iterative novelty iterations. Our novelty iteration method enhances ideas overall; however ideas are often based on superficial recombinations of common concepts, far from the technical depth of scientific papers.

this, we conduct an additional initial experiment in the biochemical domain. We follow a similar data creation procedure as for NLP papers. We collect a dataset from PubMed papers and use PubTator 3 (Islamaj et al., 2021; Wei et al., 2022; Luo et al., 2023; Wei et al., 2023; Lai et al., 2023) as an IE system to extract a KG from paper abstracts. We use a sentence classifier trained on annotated abstracts (Huang et al., 2020) to select background context. We fine-tune a state-of-the-art biomedical large language model (Chen et al., 2023) on our data and evaluate on a test split past its pre-training cutoff date.[8] We ask two biochemical domain experts with graduate-level education to evaluate the quality of the results as before, finding them to overall rate 80% of the generated directions positively. Finally, in contrast to NLP-domain experiments, evaluators were more satisfied with the generated outputs than the ground truth regarding technical detail. Detailed results are in Table 4. However, this preliminary experiment was meant mainly to demonstrate the generality of our approach, and a more in-depth exploration of utility and quality is left for future work.

4.2 Error Analysis

Models often made generic suggestions, woven together with specific details copied directly from the context (e.g., "*NLP with ML algorithms and sentiment analysis*" for some problem X, or "*data augmentation and transfer learning*" for Y, or "*BERT or RoBERTa*" for Z). Our techniques reduced this behavior but did not fully solve it. GPT4 models, especially, seemed to generate generic descriptions of common steps in NLP workflows (e.g., "*Data preprocessing: Clean the text data, remove unnecessary characters, perform tokenization...*"). All models often copied and rephrased directly from the context. In certain cases, models applied simple logical modifications to the context; e.g., when contexts described problems such as "*high latency*" or "*efficiency limitations*", the suggestions would include phrases such as "*low latency*" or "*highly efficient*".

4.3 Automated Evaluation Analysis

In open-ended tasks such as ours, automatic evaluations comparing system output to ground truth texts may be limited. Nonetheless, automated metrics such as ROUGE (Lin, 2004), BERTScore (Zhang* et al., 2020) and BARTScore (Yuan et al., 2021), that check the similarity between ground truth and generated output, may surface interesting findings. We find `GPT`-based models to be outperformed by `T5`-based models; `GPT4` outputs are much longer than `T5`, explaining why they underperform in automatic metrics but outperform in human evaluations (§4.1). Generated sentences often follow certain templates (e.g., *"In this paper, we propose a new ... for ..."*), which also helps explain why `T5` fine-tuned on many examples scores higher superficially. At the same time, our in-context contrastive examples which encourage novelty with respect to background context, helped models perform better than baseline fine-tuning by reducing reliance on copying. See results in Table 9 (Appendix B.4).

5 Conclusions and Future Directions

We propose a new setting, model and comprehensive evaluation for scientific hypothesis generation with language models that are grounded in literature and optimized for novelty. We present a new framework named SCIMON in which mod-

[8]More data and training details in Appendix A.2, B.2.3.

els take background problem contexts and provide suggestions that are novel while based on literature. Models retrieve inspirations from semantic similarity graphs, knowledge graphs, and citation networks. We introduce a new iterative novelty boosting mechanism that helps large language models (LLMs) such as GPT-4 generate more novel ideas by explicitly comparing ideas to prior work and refining them. Our experiments demonstrate that the task of generating natural language scientific hypotheses is highly challenging. While our methods improve upon baseline LLMs, generated ideas tend to be incremental and with insufficient detail. Generating novel and meaningful scientific concepts and their compositions remains a fundamental problem (Hope et al., 2023). Evaluation in this setting is also highly challenging, with a huge space of potentially plausible hypotheses formulated in natural language. One interesting direction is to expand SciMON with a multimodal analysis of formulas, tables, and figures to provide a more comprehensive background context.

6 Limitations

We discuss limitations extensively throughout the paper, such as in terms of evaluation challenges and data quality. Here we include additional details on limitations.

6.1 Limitations of Data Collection

We crawled papers with Semantic Scholar Academic Graph API from 1952 to June 2022. The number of available papers is limited by the data we crawled from the Semantic Scholar Academic Graph. We also crawled papers from PubMed 1988 to 2024/01. We remove papers that are not English. We also remove papers where abstracts are not correctly parsed from paper PDFs. We will expand our models to papers written in other languages and other domains in the future.

6.2 Limitations of System Performance

Our dataset is based on state-of-the-art IE systems, which may be noisy. For instance, the coreference and SciSpacy abbreviation resolution models fail to link *A2LCTC* to *Action-to-Language Connectionist Temporal Classification*. The background context detection may also have errors: e.g., the sentence classification component fails to treat *"For example, the language models are overall more positive towards the stock market, but there are significant differences in preferences between a pair of industry sectors, or even within a sector."* as background context. In our human-vetted gold data subset, we make sure to filter such cases, but they remain in the training data. SentenceBert (Reimers and Gurevych, 2019), and GPT3.5/4 are not finetuned and might be biased towards pretraining datasets. The idea novelty boosting method is limited by the quality of retrieval models. Better retrieval models may be explored in the future. Due to hardware constraints, we mainly investigated models with up to 7 billion parameters. Due to API change and model randomness, our GPT3.5/4 results might not be easily reproducible.

6.3 Limitations of Evaluation

We recruit annotators from Ph.D. students; their opinions may differ from annotators who have different levels of domain knowledge. Our setting uses a seed term taken from the ground truth as input, to emulate a scenario where a human provides guidance to an assistant model. Future work could explore methods in the setting without a seed term, an even harder task, or evaluate in an interactive setting with user-provided seed terms. In addition, while the seed is sampled from the ground truth, in our human-annotated gold subset, we make sure that in no case does the input context trivially leak the output.

6.4 Memorization Check

Carlini et al. (2023) reports that LLMs tend to memorize part of their training data, a well-known concern in evaluating current LLMs. Therefore, we examine the pretraining data of each model:

- T5: Raffel et al. (2020) shows that T5 is pretrained on C4 which was crawled from web prior to April 2019.
- GPT3.5: Based on the documentation,[9] GPT-3.5 series is pretrained on a combination of test and code from before Q4 2021.
- GPT4: OpenAI (2023) shows that the GPT-4 checkpoint we used utilizes most pertaining data before September 2021. Despite this, the pretraining and post-training data contain "a small amount" of more recent data.[10]

[9] `platform.openai.com/docs/model-index-for-researchers`

[10] See footnote 10, page 10 of OpenAI (2023).

Because we evaluate our models on papers published in 2022, the likelihood of test papers appearing in the pretraining corpora for the models is substantially reduced. We additionally performed a manual examination of GPT-4 memorization in our gold set based on 2022 ACL Anthology papers, by seeing if GPT-4 could complete information such as method names or generate text that strongly mimics the ground truth papers, and found no evidence of this occurring. The Meditron-7b (Chen et al., 2023) uses PubMed with a cut-off in August 2023, and our biochemical test set only includes PubMed papers after 2023/08.

Acknowledgements

This work is supported by the Molecule Maker Lab Institute: an AI research institute program supported by NSF under award No. 2019897, by DOE Center for Advanced Bioenergy and Bioproducts Innovation U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research under Award Number DESC0018420, by U.S. the AI Research Institutes program by National Science Foundation and the Institute of Education Sciences, Department of Education through Award No. 2229873 - AI Institute for Transforming Education for Children with Speech and Language Processing Challenges, and by AI Agriculture: the Agriculture and Food Research Initiative (AFRI) grant no. 2020-67021- 32799/project accession no.1024178 from the USDA National Institute of Food and Agriculture. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of, the National Science Foundation, the U.S. Department of Energy, and the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Nicholas Carlini. 2023. A llm assisted exploitation of ai-guardian. *Cryptography and Security Repository*, arXiv:2307.15008.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Arie Cattan, Sophie Johnson, Daniel S Weld, Ido Dagan, Iz Beltagy, Doug Downey, and Tom Hope. 2021. Scico: Hierarchical cross-document coreference for scientific concepts. In *3rd Conference on Automated Knowledge Base Construction*.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. 2023. Meditron-70b: Scaling medical pretraining for large language models. *Computation and Language Repository*, arXiv:2311.16079.

Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Dan Weld. 2019. Pretrained language models for sequential sentence classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3693–3699, Hong Kong, China. Association for Computational Linguistics.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Qian Dong, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2022. Learning when to translate for streaming speech. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 680–694.

Sam Henry and Bridget T McInnes. 2017. Literature based discovery: models, methods, and trends. *Journal of biomedical informatics*, 74:20–32.

Tom Hope, Doug Downey, Oren Etzioni, Daniel S Weld, and Eric Horvitz. 2023. A computational inflection for scientific discovery. *Communications of the ACM*.

Zhe Hu, Zuohui Fu, Yu Yin, and Gerard de Melo. 2021. Context-aware interaction network for question matching. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3846–3853, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2023. Benchmarking large language models as ai research agents. *Machine Learning Repository*, arXiv:2310.03302.

Ting-Hao Kenneth Huang, Chieh-Yang Huang, Chien-Kuang Cornelia Ding, Yen-Chia Hsu, and C. Lee Giles. 2020. CODA-19: Using a non-expert crowd to annotate research aspects on 10,000+ abstracts in the COVID-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

Rezarta Islamaj, Robert Leaman, Sun Kim, Dongseop Kwon, Chih-Hsuan Wei, Donald C. Comeau, Yifan Peng, David Cissel, Cathleen Coss, Carol Fisher, Rob Guzman, Preeti Gokal Kochar, Stella Koppel, Dorothy Trinh, Keiko Sekiya, Janice Ward, Deborah Whitman, Susan Schmidt, and Zhiyong Lu. 2021. Nlm-chem, a new resource for chemical entity recognition in pubmed full text literature. *Scientific Data*, 8(1):91.

Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about GPT-3 in-context learning for biomedical IE? think again. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4497–4512, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Po-Ting Lai, Chih-Hsuan Wei, Ling Luo, Qingyu Chen, and Zhiyong Lu. 2023. Biorex: Improving biomedical relation extraction by leveraging heterogeneous datasets. *Journal of Biomedical Informatics*, 146:104487.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Ling Luo, Chih-Hsuan Wei, Po-Ting Lai, Robert Leaman, Qingyu Chen, and Zhiyong Lu. 2023. AIONER: all-in-one scheme-based biomedical named entity recognition using deep learning. *Bioinformatics*, 39(5):btad310.

Rahul Nadkarni, David Wadden, Iz Beltagy, Noah A Smith, Hannaneh Hajishirzi, and Tom Hope. 2021. Scientific language models for biomedical knowledge base completion: an empirical study. *AKBC*.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

Allen Newell and Herbert Simon. 1956. The logic theory machine–a complex information processing system. *IRE Transactions on information theory*, 2(3):61–79.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023. Capabilities of GPT-4 on medical challenge problems. *Computation and Language Repository*, arXiv:2303.13375.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *Machine Learning Repository*, arXiv:1807.03748.

OpenAI. 2023. Gpt-4 technical report. *Computation and Language Repository*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Yujia Qin, Jiajie Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. ELLE: Efficient lifelong pre-training for emerging data. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2789–2810, Dublin, Ireland. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Herbert A Simon. 1973. Does scientific discovery have a logic? *Philosophy of science*, 40(4):471–480.

Daniel N Sosa and Russ B Altman. 2022. Contexts and contradictions: a roadmap for computational drug repurposing with knowledge inference. *Briefings in Bioinformatics*, 23(4):bbac268.

Don R Swanson. 1986. Undiscovered public knowledge. *The Library Quarterly*, 56(2):103–118.

Justin Sybrandt, Ilya Tyagin, Michael Shtutman, and Ilya Safro. 2020. Agatha: automatic graph mining and transformer based hypothesis generation approach. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2757–2764.

Vahe Tshitoyan, John Dagdelen, Leigh Weston, Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A Persson, Gerbrand Ceder, and Anubhav Jain. 2019. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763):95–98.

Qingyun Wang, Lifu Huang, Zhiying Jiang, Kevin Knight, Heng Ji, Mohit Bansal, and Yi Luan. 2019. PaperRobot: Incremental draft generation of scientific ideas. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1980–1991, Florence, Italy. Association for Computational Linguistics.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, et al. 2023. Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34.

Chih-Hsuan Wei, Alexis Allot, Kevin Riehle, Aleksandar Milosavljevic, and Zhiyong Lu. 2022. tmvar 3.0: an improved variant concept recognition and normalization tool. *Bioinformatics*, 38(18):4449–4451.

Chih-Hsuan Wei, Ling Luo, Rezarta Islamaj, Po-Ting Lai, and Zhiyong Lu. 2023. GNorm2: an improved gene name recognition and normalization system. *Bioinformatics*, 39(10):btad599.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yi Xu, Shuqian Sheng, Bo Xue, Luoyi Fu, Xinbing Wang, and Chenghu Zhou. 2023. Exploring and verbalizing academic ideas by concept co-occurrence. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13001–13027, Toronto, Canada. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917, Dublin, Ireland. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *Proceedings of the 8th International Conference on Learning Representations*.

Ran Zhou, Xin Li, Ruidan He, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022. MELM: Data augmentation with masked entity language modeling for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2251–2262, Dublin, Ireland. Association for Computational Linguistics.

A Dataset Collection

A.1 NLP Dataset Collection

We download ACL Anthology papers from 1952 to 2022 using Semantic Scholar Academic Graph API.[11] We filter out papers without abstracts and not written in English to obtain 67,408 papers. Our dataset has 58,874 papers before 2021, 5,946 papers from 2021, and 2,588 from 2022. We first use PL-Marker (Ye et al., 2022) pretrained on SciERC (Luan et al., 2018) to extract nodes belonging to six types: *Task*, *Method*, *Evaluation Metric*, *Material*, *Other Scientific Terms*, and *Generic*

[11] www.semanticscholar.org/product/api

Terms. The model then predicts relations between nodes belonging to seven relation types: *Used-for*, *Feature-of*, *Evaluate-for*, *Hyponym-of*, *Part-of*, *Compare*, and *Conjunction*. Because we want to generate new ideas, we focus on *used-for* relations in papers. Next, we use SciCo (Cattan et al., 2021) with checkpoint from Hugging Face[12] to obtain entity coreference to merge identical nodes. Then, we use ScispaCy (Neumann et al., 2019) to perform unsupervised abbreviation detection to replace the abbreviation with a more informative long form. Finally, we perform scientific sentence classification (Cohan et al., 2019)[13] to classify sentences from the abstract into five categories including *Background*, *Method*, *Objective*, *Other*, and *Result*. We select sentences with labels of *Background* and *Other* as background context. During preprocessing, we only keep high-confidence outputs from IE models. Figure 4 shows an example of the IE systems pipeline.

A.2 Biochemical Dataset Collection

We collect PubMed papers from 1988 to 2024 using Entrez Programming Utilities API[14] for the following topics, including *Yarrowia*, *Saccharomyces cerevisiae*, *Issatchenkia orientalis*, and *Rhodosporidium toruloides*. We use PubTator 3 (Islamaj et al., 2021; Wei et al., 2022; Luo et al., 2023; Wei et al., 2023; Lai et al., 2023). The PubTator 3 performs named entity recognition, relation extraction, entity coreference and linking, and entity normalization for the abstracts in the dataset. PubTator 3 identifies bio entities belonging to seven types: *gene*, *chemical*, *chromosome*, *cell line*, *variant*, *disease*, and *species*l and relations belonging to 13 types: *associate*, *cause*, *compare*, *convert*, *contract*, *drug interact*, *inhibit*, *interact*, *negative correlate*, *positive correlate*, *prevent*, *stimulate*, and *treat*. Finally, we use a sentence classifier trained on CODA-19 (Huang et al., 2020) to classify sentences in abstracts into *background*, *purpose*, *method*, *finding*, and *other*. We select sentences with labels of *background* as background context and remove sentences with labels of *other*. We treat the rest sentences that have at least one entity as the target sentence. We only keep samples with low similarity between background context and corresponding ground truth sentences.[15] Our final dataset has 4,767 papers before 2023/02, 642 papers from 2023/02 to 2023/08, and 299 papers after 2023/08.

[12] huggingface.co/allenai/longformer-scico
[13] github.com/allenai/sequential_sentence_classification
[14] www.ncbi.nlm.nih.gov/books/NBK25501/

B Finetuning and Automated Evaluation details

B.1 Inspiration Retrieval Module

The statistics of each inspiration type are in Table 7. Table 8 shows sample retrieved inspirations.

B.1.1 Semantic Neighbors

We use `all-mpnet-base-v2` from SentenceBert (Reimers and Gurevych, 2019), which performs best in semantic search to retrieve similar nodes from the training set based on query q in §3.1. We retrieve up to 20 relevant semantic neighbors $\mathcal{R}$ from the training set for each instance. We treat the target nodes from $\mathcal{R}$ as semantic neighbors.

B.1.2 KG Neighbors

We use one-hop connected neighbors from the background KG $\mathcal{G}_B$ constructed on papers before 2021(i.e., the papers in the training set). Because of the scarcity of KG neighbors, we do not limit the number of KG neighbors.

B.1.3 Citation Neighbors

Similar to semantic neighbors, we use $\mathtt{all-mpnet-base-v2}$ from SentenceBert (Reimers and Gurevych, 2019) to retrieve cited paper titles similar to query q. We restrict cited papers only before 2021. We retrieve up to 5 relevant citation neighbors from the papers' citation network.

B.2 Generation Module

Our T5 model and their variants are built based on the Huggingface framework (Wolf et al., 2020).[16] We optimize those models by AdamW (Loshchilov and Hutter, 2019) with the linear warmup scheduler.[17] Those models are finetuned on 4 NVIDIA A6000 48GB GPUs with distributed data parallel.[18] The training time for each model is about 10 hours.

[15] The similarity is calculated with `all-mpnet-base-v2`.
[16] github.com/huggingface/transformers
[17] huggingface.co/docs/transformers/main_classes/optimizer_schedules#transformers.get_linear_schedule_with_warmup
[18] pytorch.org/tutorials/intermediate/ddp_tutorial.html

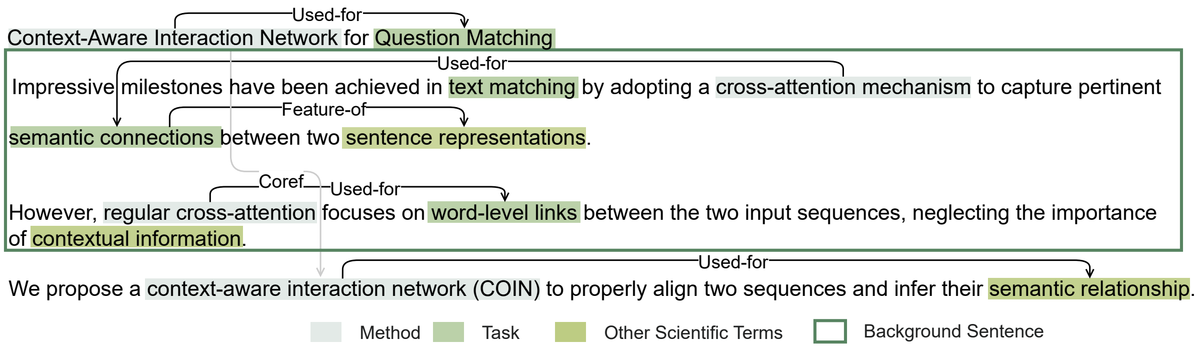

Figure 4: Preprocessing result for Hu et al. (2021) in non-canonicalized KG Corpus

Stage	PL-Maker Entities	PL-Maker Used-for Relations	SciCo Coreference	Scispacy Abbreviation Detection	Sentence Classification
Precision	91.3	65.4	97.2	100	100

Table 6: Human quality evaluation of preprocessing stages(%). Overall pass rate after all steps are applied is 79.7%.

Type	Train	Valid	Test
SN	10.8	10.0	10.0
KG	8.3	8.0	8.1
CT	4.9	5.0	5.0

Table 7: Average of # of neighbors for each instance, excluding those which do not have any neighbor

B.2.1 In-Context Learning

We choose GPT3.5 `davinci-003`[19] (Brown et al., 2020) as our out-of-the-box causal language modeling baseline. We select 5 instances from the training set as examples for the few-shot setting. We randomly select those examples for `GPT3.5FS`. For `GPT3.5Retr`, similar to semantic neighbors, we use `all-mpnet-base-v2` from SentenceBert (Reimers and Gurevych, 2019), which performs best in semantic search to retrieve similar instances from the training set based on query q in §3.1. The input length is limited to 2048 tokens due to OpenAI API limits. We choose `gpt-4-0314` as our GPT4 model. Our input for GPT4 is similar to GPT3.5.

For each selected example from the training set with *forward* relation, the template is *"Consider the following context: $\mathcal{M}$ In that context, which p can be used for v, and why? $\mathcal{T}$"*, where $\mathcal{M}$ is the background context, p is the target node type, v is the seed term, and $\mathcal{T}$ is the target idea sentence; for *backward* relation, the template is *"Consider the following context: $\mathcal{M}$ In that context, which p do we use v, and why? s"*. For selected examples with additional retrieval inspirations, we concatenate the following additional template to the $\mathcal{M}$: *"The retrieval results are: $i_1, \ldots, i_k$"*, where $i_1, \ldots, i_k$ are retrieved inspirations. For the final prompt, the template is similar to the above example template. However, the target sentence $\mathcal{T}$ will not be included. We ask the model to generate 10 outputs. We will select the best output and skip the empty output.

[19] openai.com/api/

B.2.2 Fine Tuning

Given input without any inspirations, the input combines the prompt $\mathcal{P}$ and context $\mathcal{M}$ as shown in §3.1 (i.e., $\mathcal{P}$ | *context:* $\mathcal{M}$). Given input with inspirations, the input is $\mathcal{P}$ | *retrieve:* $i_1, \ldots, i_k$ | *context:* $\mathcal{M}$, with $i_1, \ldots, i_k$ as retrieved inspirations. The input length is limited to 512 tokens. For both tasks, we finetune our model based on `T5-large` with a learning rate of 6×10^{-6} and $\epsilon = 1 \times 10^{-6}$. The batch size is 8 for each GPU. The maximum training epoch for all models is 10 with 4 patience. During decoding, we use beam-search to generate results with a beam size of 5 and a repetition penalty of 1.5.

In-context Contrastive Augmentation We randomly select 2 sentences that appeared in the input as in-context negatives. For example, in Figure 1, the in-context negatives could be *"knowledge acquisition is done by using Method"*, *"this requires plms to integrate the information from all the sources in a lifelong manner ."*.

B.2.3 Biochemical Case Study

Our Meditron-7b (Chen et al., 2023) and its variants are built based on the Huggingface framework (Wolf et al., 2020).[20] We use its `epfl-llm/meditron-7b` as the base model. We finetune those models with a learning rate of 2×10^{-6} and $\epsilon = 5 \times 10^{-8}$. The maximum training epoch for all models is 5. All models are fine-tuned on 4 NVIDIA A100 80 GB GPUs with Fully Sharded Data Parallel.[21] The training time for each model is about 20 hours.

B.3 The Scale of Retrieval Set

We retrieve from a set of 59k papers with over 374k sentences in the NLP domain, the focus of our experiments. Our background KG built on the training set has more than 197k nodes and 261k relations. Moreover, we collect 87k paper titles from citation networks. This represents a large-scale and diverse domain; retrieving inspirations from this set is expected, in principle, to be more than enough for generating novel ideas. Indeed, NLP papers typically cite each other and build on each other as inspirations to create new ideas - which motivates our inspiration retrieval.

B.4 Automated Evaluation

We use BERTScore (Zhang* et al., 2020) with SciBERT checkpoint for both tasks. The hash of the checkpoint is `allenai/scibert_scivocab_uncased_L8_no-idf_version=0.3.12(hug_trans=4.19.2)`. The automated evaluation results are in Table 9.

C Human Annotation and Evaluation Details

Gold Dataset Annotation Details The gold dataset annotation interface is in Figure 5. The quality of the instances in the test set is judged given three criteria: (1) whether the ground truth sentence trivially overlaps with background context; (2) whether background context contains relevant information for the target relation; (3) whether the target relation (from which the seed term is taken) is a salient aspect of the idea proposed in the target paper.

Study I The instructions for human evaluation can be found in Figure 6, while an example of the human evaluation interface is provided in Figure 7 and 8. Human annotators are required to evaluate each system output based on the following criteria: (1) *Is the candidate relevant to the context + seed term?* (2) *Does the candidate copy too much from the context, or is it sufficiently novel/different from the context?* (3) *Does the candidate's suggestion generally make sense to you scientifically?* (4) *Is the language sufficiently clear and coherent to understand the suggestion?* The input for sample human annotation is in Table 10 and the human labels are in Table 11. The human annotation agreement is in Table 13.

Study III We ask the following questions to human annotators to evaluate the quality of regeneration results: (1) *Is the regenerated idea substantially different from the original?* (2) *Is the regenerated idea more novel and creative than the original idea?* (3) *Does the second iteration increase novelty?* The human annotation agreement is in Table 14.

D Scientific Artifacts

We list the licenses of the scientific artifacts used in this paper: Semantic Scholar Academic Graph API (API license agreement[22]), Huggingface Transformers (Apache License 2.0), SBERT (Apache-2.0 license), BERTScore (MIT license), Meditron-7b (Llama2), Entrez Programming Utilities API (Copyright[23]), PubTator 3 (Data use policy[24]), and OpenAI (Terms of use[25]).

E Ethical Consideration

The SciMON task and corresponding models we have designed in this paper are limited to the natural language processing (NLP) and biochemical domain, and might not apply to other scenarios.

E.1 Usage Requirement

This paper aims to provide investigative leads for a scientific domain, specifically natural language processing. The final results are not intended to be used without human review. Accordingly, domain experts might use this tool as a research writing assistant to develop ideas. However, our system does not do any fact-checking with external knowledge. In addition, we train our models on the ACL

[20] `github.com/huggingface/transformers`

[21] `https://huggingface.co/docs/accelerate/usage_guides/fsdp`

[22] `api.semanticscholar.org/license/`

[23] `www.ncbi.nlm.nih.gov/books/about/copyright/`

[24] `www.ncbi.nlm.nih.gov/home/about/policies/`

[25] `openai.com/policies/terms-of-use`

Type	Content
Seed Term Prompt	data augmentation is used for Task
Context	data augmentation is an effective solution to data scarcity in low - resource scenarios. however, when applied to token-level tasks such as ner , data augmentation methods often suffer from token-label misalignment, which leads to unsatsifactory performance.
Semantic Neighbors	st and automatic speech recognition (asr), low-resource tagging tasks, end-to-end speech translation, neural online chats response selection, neural machine translation, semi-supervised ner, entity and context learning, semi-supervised setting, dependency parsing, low-resource machine translation, slot filling, dialog state tracking, visual question answering, visual question answering (vqa), low-resource neural machine translation
KG Neighbors	nmt-based text normalization, task-oriented dialog systems, task-oriented dialogue system, low-resource languages (lrl), end-to-end speech translation, visual question answering (vqa), multiclass utterance classification, clinical semantic textual similarity, neural online chats response selection, context-aware neural machine translation
Citation Neighbors	Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations, An Analysis of Simple Data Augmentation for Named Entity Recognition, Data Augmentation for Low-Resource Neural Machine Translation, DAGA: Data Augmentation with a Generation Approach for Low-resource Tagging Tasks, EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks
Ground Truth	ELM: Data Augmentation with Masked Entity Language Modeling for Low-Resource NER

Table 8: Example (from (Zhou et al., 2022)) of retrieved inspirations. Inspirations similar to ground truth are underlined.

input	context	entity	output	relation	rel_sent	Is the output trivally overlap with the context	IE is of sufficient quality (not generic, correct)	context contains relevant information for target relation (Conservative filter - only flag cases where context is highly irrelevant)	Relation is a part of the main idea proposed by the paper
extractive text summarization is done by using Metric	transformer - based language models usually treat texts as linear sequences . however , most texts also have an inherent hierarchical structure , i.e. , parts of a text can be identified using their position in this hierarchy . in addition , section titles usually indicate the common topic of their respective sentences .	extractive text summarization	sota rouges	used for	We propose a novel approach to formulate , extract , encode and inject hierarchical structure information explicitly into an extractive summarization model based on a pre - trained , encoder - only Transformer language model (HiStruct+ model) , which improves SOTA ROUGEs for extractive summarization on PubMed and arXiv substantially .				

Figure 5: Gold subset annotation interface

anthology and PubMed papers written in English, which might alienate readers who have been historically underrepresented in the NLP/biochemical domains.

E.2 Data Collection

We collect 67,408 ACL Anthology papers from 1952 to 2022 using Semantic Scholar Academic Graph API, under API license agreement[26]. We ensure our data collection procedure follows the Terms of Use at https://allenai.org/terms. According to the agreement, our dataset can only be used for non-commercial purposes. As mentioned in §4, we perform the human evaluation. All annotators involved in human evaluation are voluntary participants with a fair wage. We further collect 5,708 PubMed papers from 1988 to 2024 using Entrez Programming Utilities API[27]. We follow their data usage guidelines[28].

[26] https://api.semanticscholar.org/license/

[27] www.ncbi.nlm.nih.gov/books/NBK25501/

[28] www.ncbi.nlm.nih.gov/books/about/copyright/

Subset	**Challenging**		**Gold**	
Model	**R-L↑**	**BERT↑**	**R-L↑**	**BERT↑**
GPT4ZS	0.120	0.581	0.130	0.583
GPT4FS	0.143	0.618	0.151	0.624
T5	0.223	**0.672**†	0.246	0.685
GPT4FS+SN	0.144	0.620	0.149	0.627
GPT4FS+KG	0.143	0.619	0.152	0.626
GPT4FS+CT	0.144	0.617	0.149	0.622
T5+CL	0.225†	0.671†	0.251†	0.686†
T5+SN+CL	**0.228**†	0.671†	**0.258**†	0.686†
T5+KG+CL	0.223†	0.669	0.248	0.681†
T5+CT+CL	0.225†	0.671†	0.250†	**0.686**†

Table 9: Automatic evaluation results for the challenging and gold subsets. *CL* is a model with in-context contrastive augmentation. *SN* is a model with semantic inspirations. *KG* is a model with KG inspirations. *CT* is a model with citation inspirations. *R-L* denotes ROUGE-L. *BERT* denotes BERTscore with SciBERT as its encoder. † indicates that differences between models are not statistically significant ($p \leq 0.05$) when compared to each other but are still significant when compared to the other models based on t-test.

Type	Content
Seed Term Prompt	Irish language learning is done by using Method
Context	Irish is a minority language which means that l2 learners have limited opportunities for exposure to the language, and additionally, there are also limited (digital) learning resources available.

Table 10: Input for sample human annotation results

Model	Output	Label
GPT3.5FS	The use of Social Media in **Irish Language Learning**: A Case Study	*helpful*
GPT3.5Retr	One method that could be used for **Irish language learning** is computer assisted language learning (CALL)	*unhelpful*
GPT3.5FS+CT	In that context, the most appropriate task for precision grammar would be the development of a part-of-speech tagger for Irish using finite-state morphology and constraint grammar disambiguation.	*unhelpful*
GPT3.5FS+KG	Cló: An **Irish Language Learning** Platform - Leveraging Digital Resources and Human Expertise	*unhelpful*
GPT4FS	We propose using a technology-enhanced approach, such as computer-assisted language learning (CALL) or natural language processing (NLP) techniques, to create interactive and engaging **Irish language learning** resources, as these methods can provide personalized learning experiences and enhance learners' exposure to the language despite limited resources available.	*helpful*
GPT4FS+KG	Exploiting Social Media for Irish Language Learning: An Analysis of Twitter Data. In this context, we use social media data, particularly from Twitter, as a method for **Irish language learning**, because it provides a rich source of authentic and diverse language examples that can be used to enhance learning opportunities for L2 learners in a minority language setting.	*helpful*
T5	In this paper, we present a novel machine translation approach to **Irish language learning**.	*unhelpful*
T5+SN+CL	A conversational virtual environment for **Irish language learning**	*helpful*

Table 11: Sample human labels for idea-sentence generation results for seed term prompt ***"Irish language learning** is done by using Method"*, with context *"irish is a minority language ... limited digital learning resources available"*.

Type	Content
Seed Term	ARO10
Context	Huangjiu is known for its unique aroma, primarily attributed to its high concentration of b-phenylethanol (ranging from 40 to 130 mg/L).
Meditron	We then selected six key genes involved in the phenylalanine metabolism pathway and predicted that their product flux may affect the synthesis of b-phenylethanol.
Meditron + CT	We found that the key amino acid residue that controls the activity of **Aro10p** was not conserved in wine yeast strains, which may explain the lower b-phenylethanol production in wine fermentation compared with that in Chinese huangjiu.
Meditron + KG	Both target genes, SSA1 and **ARO10**, were deleted using the CRISPR-Cas9 genome editing system.
Meditron + SN	Herein, we report that the key barrier for b-phenylethanol production in Huangjiu is **ARO10**, the only bi-functional amino acid decarboxylase in Saccharomyces cerevisiae.

Table 12: Input and idea-sentence generation results for seed gene "***ARO10***" in the biochemical domain.

Annotator Pair	1-2	1-3	1-4	1-5	1-6
Agreement %	68.8	75.0	56.2	43.8	75.0

Table 13: Percent (%) of same labels from overlapped 10 human evaluation instances on each pair of annotators for Study I.

Annotator Pair	1-2	1-3	1-4	1-5
Agreement %	92.5	93.3	87.5	90.0

Table 14: Percent (%) of same labels from overlapped 20 human evaluation instances on each pair of annotators for Study III. (1-3) has 60 shared questions. The rest of the pairs each share 40 questions.

Rank scientific idea suggestions generated by an AI paper-writing assistant

Your goal in this task is to rank idea suggestions written by an AI assistant. The AI assistant helps its users write paper abstracts by writing sentences with proposals for new ideas or questions to consider.
You are first given:

1. A context which describes relevant background in a specific area of interest. *e.g., background on efficiency issues in large language model training.*
2. A seed term that should be a focus of the generated scientific idea. see example below.

Then, you are given a list of 5 candidate suggestions — each generated by the AI assistant using different approaches. Each candidate is written in the form of a sentence from a scientific paper abstract.

Your goal is to score the candidate suggestions with a helpful/unhelpful answer. Your scoring decision should consider the overall quality of each candidate, using considerations that include:

1. Is the candidate relevant to the context + seed term?
2. Does the candidate copy too much from the context, or is it sufficiently novel/different from the context?
3. Does the candidate's suggestion generally make sense to you scientifically?
4. Is the language sufficiently clear and coherent for you to understand the suggestion?

The AI scientific assistant is a work in progress – don't expect it to generate highly-detailed ideas or breakthrough discoveries! Use the three guideline metrics above to come up with your overall rating.
Read the following example that illustrates the task:

Context: *the task of converting a natural language question into an executable sql query , known as text - to - sql , is an important branch of semantic parsing . the state - of - the - art graph - based encoder has been successfully used in this task but does not model the question syntax well .*
Seed term: *diverse relational edge embedding*
System Outputs:

- *We propose a novel technique , Diverse Relational Edge Embedding (DREE) , to address this problem.*
- *We propose a novel graph-based encoder that uses a diverse relational edge embedding to model the question syntax.*
- *Diverse relational edge embedding is important for text-to-SQL parsing because it can help the parser to better understand the question syntax.*
- *In this paper, we propose a novel graph - based encoder for diverse relational edge embedding.*
- *In this paper, we propose a novel graph - based encoder for text - to - sql (ttsql) that incorporates the question syntax into diverse relational edge embedding.*

Ranking:

- unhelpful
- helpful
- helpful
- unhelpful
- helpful

Figure 6: Human evaluation instructions

Your goal is to score the candidate suggestions with a helpful/unhelpful answer. Your scoring decision should consider the overall quality of each candidate, using considerations that include: *

Is the candidate relevant to the context + seed term?

Does the candidate copy too much from the context, or is it sufficiently novel/different from the context?

Does the candidate's suggestion generally make sense to you scientifically?

Is the language sufficiently clear and coherent for you to understand the suggestion?

Context: *while pretrained language models achieve excellent performance on natural language understanding benchmarks , they tend to rely on spurious correlations and generalize poorly to out - of - distribution (ood) data . recent work has explored using counterfactually - augmented data (cad)-data generated by minimally perturbing examples toflip the ground - truth label - to identify robust features that are invariant under distribution shift .*

Seed term: *diverse perturbation of examples*

	unhelpful	helpful
Diverse perturbation of examples is used in order to generate counterfactual data that can help identify robust features that are invariant under distribution shift.	○	○
A counterfactual generator for diverse perturbation of examples.	○	○
We propose a method for generating CAD by diverse perturbation of examples.	○	○
In this paper, we propose a counterfactually - augmented data (cad) model that is robust to diverse perturbation of examples.	○	○
We use diverse perturbation of examples to flip the ground-truth label in order to identify robust features that are invariant under distribution shift.	○	○

Figure 7: Human evaluation example for GPT3.5Rnd, GPT3.5Retr, GPT3.5Rnd+CT, T5, and T5+SN+CL

Your goal is to score the candidate suggestions with a helpful/unhelpful answer. Your scoring decision should consider the overall quality of each candidate, using considerations that include: *

Is the candidate relevant to the context + seed term?

Does the candidate copy too much from the context, or is it sufficiently novel/different from the context?

Does the candidate's suggestion generally make sense to you scientifically?

Is the language sufficiently clear and coherent for you to understand the suggestion?

Context: *while pretrained language models achieve excellent performance on natural language understanding benchmarks , they tend to rely on spurious correlations and generalize poorly to out - of - distribution (ood) data . recent work has explored using counterfactually - augmented data (cad)-data generated by minimally perturbing examples toflip the ground - truth label - to identify robust features that are invariant under distribution shift .*
Seed term: *diverse perturbation of examples*

	unhelpful	helpful
In that context, we use diverse perturbation of examples to generate counterfactually-augmented data, which helps identify robust features and improve the model's generalization to out-of-distribution data. This approach minimizes the model's reliance on spurious correlations and enhances its overall performance on natural language understanding tasks.	○	○
In that context, we use diverse perturbation of examples as a method because it helps in generating counterfactually-augmented data (CAD), which helps identify robust features that are invariant under distribution shift. By focusing on robust features, the model becomes less reliant on spurious correlations and can generalize better to out-of-distribution (OOD) data.	○	○
We propose a general framework based on diverse perturbation of examples that can be used to learn invariant features powerful enough to detect OOD data and reduce generalization error while performing natural language understanding tasks.	○	○

Figure 8: Human evaluation example for GPT3.5Rnd+KG, GPT4Rnd, and GPT4Rnd+KG