Learning to Generate Novel Scientific Directions with Contextualized Literature-based Discovery

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

Literature-Based Discovery (LBD) aims to discover new scientific knowledge by mining papers and generating hypotheses. Standard LBD is limited to predicting pairwise relations between discrete concepts (e.g., drug-disease links), and ignores critical contexts like experimental settings (e.g., a specific patient population where a drug is evaluated) and background motivations (e.g., to find drugs without specific side effects). We address these limitations with a novel formulation of *contextualized*-LBD (C-LBD): generating scientific hypotheses in *natural language*, while grounding them in a context that controls the hypothesis search space. We present a modeling framework using retrieval of "inspirations" from past scientific papers. Our evaluations reveal that GPT-4 tends to generate ideas with overall low technical depth and novelty, while our inspiration prompting approaches partially mitigate this issue. Our work represents a first step toward building language models that generate new ideas derived from scientific literature. [1]

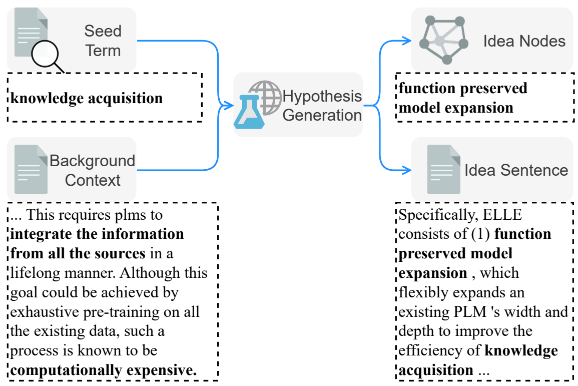

Figure 1: Contextualized Literature-Based Discovery (C-LBD) overview. In C-LBD, the system is given a description of background knowledge and a seed term to generate a new idea. Example from Qin et al. (2022).

1 Introduction

Can machines mine scientific papers and learn to suggest new directions? 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 focuses on hypothesizing links between previously unstudied pairs of concepts (e.g., new drug-disease links), with a primary application area of drug discovery (Henry and McInnes, 2017; Sybrandt et al., 2020).

While LBD systems have evolved from Swanson's simple keyword co-occurrence analyses into machine learning approaches (Sybrandt et al., 2020), the basic premise remains: predicting pairwise links between concepts from papers. This 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. Science is done within specific nuanced settings important for qualifying claims mined from literature (Sosa and Altman, 2022) — e.g., a drug may treat a disease only when administered in a certain way for certain types of patients. Such nuances are difficult to capture with information extraction (IE) systems (Luan et al., 2018; Lin et al., 2020; Ye et al., 2022) that require pre-specified schema and are limited in precision and coverage, and are largely ignored in LBD (Sosa and Altman, 2022). More broadly, LBD does not model *contexts* that human scientists consider in the ideation process (Hope et al., 2023): target application settings, requirements and constraints, motivations and challenges. Finally, the transductive LBD setting, where all concepts are known apriori and only need to be connected to each other, fails to account for the *inductive* and *generative* nature of science where new concepts and their recombinations continuously emerge.

We propose *Contextual Literature-Based Discov-*

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

ery (C-LBD), a novel setting and modeling framework with a dramatically different and broader approach that addresses these fundamental limitations. Our task is designed in alignment with modern NLP research: it is the first to ground LBD in a natural language context to constrain the generation space, and it also departs from traditional LBD in the output with a sentence generation task. In C-LBD, we are motivated by imagining an AI-based assistant that suggests ideas in natural language, including novel concepts and linkages. As illustrated in Figure 1, the assistant takes as input two contexts: (1) Relevant background, e.g., current problems, motivations and constraints, and (2) a seed term that should be a focus of the generated scientific idea. Given this input, we explore two variants of C-LBD: generating a full sentence describing an idea, and also a more limited objective of generating a salient concept of the idea (see Figure 1). We present a new modeling framework for C-LBD (see Figure 2) that retrieves *inspirations* from heterogeneous sources derived from scientific literature (e.g., past problems and their solutions, context from a scientific knowledge graph) and uses them to generate new hypotheses. We also introduce an *in-context contrastive model* which encourages novelty with respect to background sentences.

We perform comprehensive evaluation of language models for generating scientific ideas. We focus on AI/NLP ideas to facilitate analysis by AI researchers themselves, and also demonstrate our approach's generalization to the biomedical domain. We design extensive evaluation experiments using not only automated metrics but also human annotators with domain expertise to assess relevance, utility, novelty, and technical depth. Our retrieval-augmented hypothesis generation methods substantially improve results; however, analyses show that ideas generated by state-of-the-art models (GPT-4 (OpenAI, 2023)) fall far behind scientific papers in terms of novelty, depth and utility (§5)—raising fundamental challenges toward building machines that generate scientific directions.

2 Contextual Literature-Based Discovery

As a brief recap of Literature-based discovery (LBD) — 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. For example, by analysis of literature using only the co-occurrences of keywords, Swanson discovered a previously unknown link between Raynaud's disease and fish oil via their shared relationship with blood viscosity, leading to a clinical trial on fish oil for treating the condition (DiGiacomo et al., 1989). More recent work has used word vectors (Tshitoyan et al., 2019) or link prediction models (Wang et al., 2019; Sybrandt et al., 2020) to discover scientific hypotheses as pairwise links between concepts.

To address the limitations discussed in the Introduction, we propose *Contextual Literature-Based Discovery* (C-LBD). Figure 1 illustrates our setting: given a background description and a seed term to focus on, our aim is to generate idea suggestions. For example, Figure 1 shows a background 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. Importantly, generated ideas should not merely paraphrase the background—the output should be novel with respect to the context. We propose two different formulations of C-LBD: *idea sentence generation* and *idea node prediction*. While both tasks utilize the same input format, they diverge in the form of output they target. We describe these task formulations, and how we can obtain large-scale training and evaluation data derived from literature.

Idea-Sentence Generation Our goal in this setting is to generate a sentence describing an idea, given sentences describing background context and a seed term to guide the generation (see Figure 1). We first use information extraction (IE) to obtain large-scale training data in this form automatically (during training each instance in our automatically-built dataset only contains a seed entity paired with background sentences and the target hypothesis; during evaluation, the target hypothesis is not visible). More formally, we are given a set of documents $\mathcal{D}$, where each document contains the title and abstract of a paper. To obtain background context and meaningful seed terms, we assume each $d \in \mathcal{D}$ is annotated with a graph $G_d = (V, E)$ obtained by information extraction systems (§A). Each node $v \in V$ is a cluster of coreferential scientific entity mentions represent-

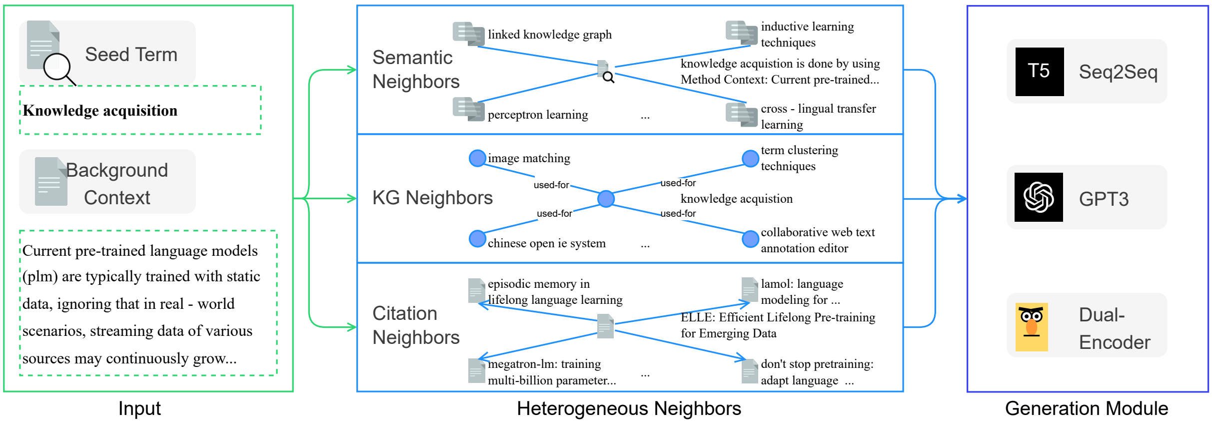

Figure 2: Architecture overview. Our models first retrieve heterogeneous *inspirations* from different sources. We then pass the contextualized input along with retrieved heterogenous neighbors to the generation modules. Finally, an in-context contrastive learning strategy is applied to reduce over-copying.

ing a concept, and each directed edge $e_{ij} \in E$ is a relation between two mentions v_i and v_j (e.g., the `[method,used-for,task]` relation). Each directed edge $e_{ij} \in E$ is then paired with a context sentence $s_{ij} \in d$—the sentence mentioning e_{ij}. Each document $d \in \mathcal{D}$ also comes annotated with a background context $\mathcal{B}$ (§A)). We treat the remaining sentences s_{ij} as target sentences containing new relationships e_{ij}. Given a node $v_i \in G_d$, background context $\mathcal{B}$ from a specific document d, and desired target node type p, our goal is to generate a full sentence s_{ij} describing a relationship between two entities v_i and v_j, where v_j may be a new node not already present in the graph $\mathcal{G}_d$. This is illustrated in Figure 1, where v_i corresponds to "knowledge acquisition", $\mathcal{B}$ to the *Background Context*, and s_{ij} is the *Description Sentence* which describes a *used-for* relation in which a *method* ("function preserved model expansion") is used for the *task* of knowledge acquisition.

Idea-Node Prediction Apart from idea sentence generation, we also formulate hypothesis generation as a link prediction task to more closely resemble the traditional hypothesis generation setting. Unlike standard work in LBD/hypothesis generation, our setting is of *contextual* prediction, where the context is provided via natural language sentences as in the above. Given the context, we predict new links between existing concepts, and generate new concepts. Our contextual node prediction is formulated using similar notations as above: given a node $v_i \in G_d$, background context $\mathcal{B}$ from a specific document d, and desired target node type p belonging to one of six types (*Task*, *Method*, *Evaluation Metric*, *Material*, *Other Scientific Terms*, *Generic Terms*), the model needs to predict a new node $v_j \in \mathcal{G}$ related to it (e.g., in Figure 1, predicting "function preserved model expansion" given the seed+context), where $\mathcal{G} = \cup_{d \in \mathcal{D}} G_d$ is a knowledge graph which covers all papers in the corpus $\mathcal{D}$, built as a union of all paper-specific graphs extracted automatically with IE as described above.

3 Models

We now present our models for the new tasks. We present a new module to retrieve heterogeneous inspirations to enrich the contextual input (§3.1). Then, we describe another module to generate a hypothesis (idea) given the enriched context (§3.2). Finally, we introduce a new in-context contrastive augmentation (§3.3). For details on the training setups and hyperparameter tuning for the models described below, see 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+seed by retrieving "inspirations"—related pieces of information from different sources that can guide the hypothesis generation.

As illustrated in Figure 2, for a given instance of the C-LBD task, our retrieval augmentation can retrieve different neighbors from three types of graphs (described below) and add them as additional input. Given the seed term v and its corresponding background context $\mathcal{B}$ from document d, we construct a string q which is the concatenation of $\mathcal{B}$ and the prompt $\mathcal{P}$ which is based on v and one of two templates: "v is used for p" or "v is done by using p".[2] p is the target node type.

For example, in Figure 2, the concatenation is "*Knowledge acquisition is done by using Method Context: Current pre-trained language models (plm) are typically trained with static data...*". We use q, v, and d as queries for each graph respectively. We concatenate each particular type of retrieved neighbor with the input in the following experiments. We now describe the three graphs we leverage: a semantic similarity graph, a knowledge graph, and a citation graph.

Semantic Neighbors Input-target pairs in the training set can serve as a guiding reference for predicting a new target given an input. To find input-target pairs that are relevant for a specific query q, we construct a fully connected graph $\mathcal{G}_S$ based on the training set, where each node is a pair of input text q_i and target entity u_i. We define the weight between two nodes i and j as the cosine similarity between q_i and q_j based on representations from an encoder such as SentenceBERT (Reimers and Gurevych, 2019). Given q, we first insert it into $\mathcal{G}_S$ and compute the weights of its connected edges. We then retrieve neighbors $\mathcal{R} = \{R_1, R_2, ..., R_k\}$ from the training set with the largest edge weight, where k is the number of retrieved instances, and R_i is a pair of input text q_i and target node u_i. We consider the target node set $\mathcal{U}_{\mathcal{D}} = \{u_1, u_2, ..., u_k\}$ as potentially relevant nodes. For example, in Figure 2, we find the string *"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 select its target *"linked knowledge graph"*. We concatenate those entities to form the relevant document context $[u_1; u_2; ...; u_k]$.

KG Neighbors In addition to using semantic relatedness based on neural text embeddings to find relevant training examples, we also explore enriching the context by linking it to a background KG with useful information on related methods and tasks. In particular, given the seed term v, we select one-hop connected neighbors denoted as $\mathcal{N}_v = \{n_1, n_2, ...\}$ from the background KG $\mathcal{G}_B$, where the background KG $\mathcal{G}_B = \cup_{d\in\mathcal{D}_\mathcal{T}} G_d$ is the KG which covers all papers in the corpus $\mathcal{D}_\mathcal{T}$ prior to a given year $\mathcal{T}$. 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 additional context. We concatenate those entities to form the KG context $[n_1; n_2; ...]$.

Citation Neighbors Another notion of contextual relatedness we explore is via citation graph links, which may capture relevant context missed by textual similarity measures. Given the query document d, we consider its cited paper title set $\mathcal{C}_d$ as potential candidates. Because the training set only contains paper before year $\mathcal{T}$, we only select papers $\mathcal{C}_{d\mathcal{T}} \subseteq \mathcal{C}_d$ prior to year $\mathcal{T}$. We then retrieve the top-k titles with the highest cosine similarity to the query q from the selected cited paper $\mathcal{C}_{d\mathcal{T}}$ based on the representation provided by the encoder. 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 paper title *"episodic memory in lifelong language learning"* as additional information. We concatenate those titles as cited paper context $[c_1; c_2; ...]$ where $c_i \in \mathcal{C}_{d\mathcal{T}}$. By retrieving from cited papers, we limit the set of candidates in the semantic graph $\mathcal{G}_S$ to a more relevant subset. This can be seen as a stronger assumption on information available to the model— assuming a researcher using the model provides background context in the form of relevant candidate documents from which ideas could be pooled.

3.2 Generation Module

Given retrieved inspirations, a generation module generates the hypothesis. We include several baselines in text generation and node prediction tasks.

Causal Language Modeling 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 (ZS) setting without any in-context examples (`GPT3.5ZS`, `GPT4ZS`). We then ask the model to generate sentences or the top-10 node predictions in a few-shot (FS) 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 exam-

[2] Other templates we experimented with had similar performance in our experiments, but future work could explore optimal prompting strategies for hypothesis generation.

ples, 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 retrieval discussed above, in that we provide both the input and output for each instance rather than solely supplying target entities as additional context.

Sequence-to-Sequence Fine Tuning We finetune T5 (Raffel et al., 2020; Ribeiro et al., 2021) (more recent models may be used too; see our biomedical experiment §5). T5 has recently been used for KG link prediction: KGT5 (Saxena et al., 2022) uses compositional entity representations and autoregressive decoding for link prediction inference to reduce model size while outperforming traditional models. We apply diverse beam search (Vijayakumar et al., 2017) for decoding to increase the predictions' diversity (`T5+Div`). We also experiment with constrained beam search (Anderson et al., 2017) (`T5+CBS`) to decrease the search space by covering only sequences observed in the bank of entities in the data (training+dev+test).

Dual-Encoder For node prediction, we explore a dual-encoder (Wang et al., 2021) which relies on encoder language models such as BERT (Devlin et al., 2019) and has recently been used in link prediction (Safavi et al., 2022). Recently, Wang et al. (2022) proposed SimKGC which uses a contrastive learning architecture to improve efficiency; we use SciBERT (Beltagy et al., 2019) and CS_RoBERTa (Gururangan et al., 2020) (`SciBERT` and `CSRoBERTa`).

3.3 In-context Contrastive Augmentation

We observe that the generation models tend to copy phrases from the background context. For example, given the context *"...hierarchical tables challenge numerical reasoning by complex hierarchical indexing..."*, the model will generate *"numerical table reasoning"* as the top prediction. For generating suggestions of *novel* ideas, we may wish to discourage overly copying from the background information. We introduce a new in-context contrastive objective, where negative examples are taken from the text in the input context (e.g., in Figure 1, the in-context negatives are *plms*, *pre-training*, etc). Specifically, for the dual-encoder model, we directly add the negatives to the existing InfoNCE loss. For the sequence-to-sequence model, we follow Wang et al. (2023) to compute an InfoNCE loss over the hidden states of the decoder, aiming to maximize the probability of the ground truth against the probabilities 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.

Dataset	Split	Forward	Backward	Total
Sentence Generation	Train	55,884	58,426	114,310
	Valid	7,938	8,257	16,195
	Test	2,623	2,686	5,309
Node Prediction	Train	57,916	60,093	118,009
	Valid	8,217	8,518	16,735
	Test	2,723	2,818	5,541

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*.

4 Experiments

4.1 Dataset

We construct our dataset from 67,408 ACL Anthology papers from S2ORC (Lo et al., 2020). Given a title and the corresponding abstract from a document d, we first apply state-of-the-art scientific IE system PL-Marker (Ye et al., 2022) to extract nodes $v \in V$ and relations which are edges $e \in E$ between nodes. Next, we use SciCo (Cattan et al., 2021) to obtain coreference links. Then, we use ScispaCy (Neumann et al., 2019) to replace abbreviations with a more informative long form. Finally, we perform scientific sentence classification (Cohan et al., 2019) to find background context $\mathcal{B}$. 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). Table 1 shows data statistics. More details are in Appendix A.

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 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%. See Table 6 in Appendix and data limitations in §7.

4.2 Idea Sentence Generation Evaluation

We begin with automatic evaluation, followed by human evaluation in §5. We use standard evaluation metrics, including ROUGE (Lin, 2004), BERTScore (Zhang* et al., 2020) and BARTScore (Yuan et al., 2021), to check the similarity between ground truth and generated output.

Subset	**Challenging**		**Gold**	
Model	**R-L↑**	**BERT↑**	**R-L↑**	**BERT↑**
GPT3.5ZS	0.136	0.624	0.147	0.631
GPT3.5FS	0.189	0.641	0.210	0.655
GPT3.5Retr	0.185	0.642	0.199	0.653
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
GPT3.5FS+SN	0.183	0.640	0.191	0.649
GPT3.5FS+KG	0.187	0.642	0.201	0.653
GPT3.5FS+CT	0.169	0.641	0.182	0.649
GPT4FS+KG	0.143	0.619	0.152	0.626
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 2: Automatic evaluation results on idea sentence generation for the challenging and gold subsets. *CL* is a model with in-context contrastive augmentation. *SN* is a model with semantic neighbors. *KG* is a model with KG neighbors. *CT* is a model with citation neighbors. *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.

We remove test instances where models can trivially use surface-level background information to infer the ground truth. We select test instances with low similarity between background context and corresponding ground truth sentences. To create this subset, we compute the cosine similarity between each instance's background context and its corresponding ground truth sentence in the test set. We then choose pairs with cosine similarity lower than 0.074, which amounts to the tenth percentile of pairs with the lowest cosine similarity.

We further annotate this subset to create a gold subset. We manually exclude instances with trivial overlap between the ground truth sentence and background context. We retain only instances where the background context is relevant to the target relation and where the target relation (from which the seed term is taken) is central to the target paper's idea. We also removed test pairs that have abbreviations or unexplained terms in the background context. Full annotation details are in Appendix G. We obtained 194 instances satisfying those conditions.[3]

In Table 2, our inspiration-enhanced models achieve superior performance. GPT-FS models are outperformed by all other T5-based models. In addition, GPT-FS (few-shot) approaches are considerably better than GPT-ZS (zero-shot).[4] We find GPT4 outputs to be much longer than GPT3.5 (§D), explaining why they underperform in automatic metrics but outperform GPT3.5 in human evaluations (§5). Finally, our in-context contrastive examples, which encourage novelty with respect to background context, help models perform better than baselines in most cases. Results from the semantic and citation graphs have better performance than with KG — many test instances with new or rare concepts do not have any KG neighbors; however, the KG can guide GPT4 to compose more concrete ideas (§5 and Table 3). The main takeaways, though, are that our task is challenging even for strong large language models, and that automatic evaluation metrics are highly problematic in our setting (e.g., ROUGE-L shows low variation across the different models partially due to the generated sentences often following certain templates (e.g. *"In this paper, we propose a new ... for ..."*); we conduct human evaluations and analysis in §5.

4.3 Idea Node Prediction Evaluation

Adding citation neighbors leads to a performance increase for SciBERT, CSRoBERTa, and T5. Generally, however, results indicate the task's highly challenging nature, both in modeling and evaluation. We present and discuss results in Appendix E. We also conduct a multi-choice evaluation, to test the models' ability to distinguish between relevant candidates and distractors (Appendix E.3).

[3]We also evaluate the full dataset without any filters, obtaining higher results as expected but with similar overall findings (see Table 9 in the Appendix).

[4]We also collected human ratings for GPT4-ZS vs. GPT4-FS using the criteria in §5; GPT4-FS ranked higher in 65% of cases, with the rest mostly tied.

Type	Content
Input	*seed*: **class imbalance** is done by using; *context* (abridged): ... class imbalance and drift can sometimes be mitigated by resampling the training data to simulate a known target distribution, but what if the target distribution is determined by unknown future events?
GPT4FS	In that context, we use resampling the training data to address **class imbalance**, as it helps simulate or compensate for a known target distribution even when dealing with unknown future events.
GPT4FS + KG	In that context, we use a method called Frustratingly Easy Model Averaging (FEDA) to handle **class imbalance**, as it combines the outputs of multiple weak learners trained on different subsets of the training data. This approach helps in mitigating the impact of **class imbalance** by leveraging the predictions of various models, thus providing a more robust and generalized decision. Additionally, FEDA can adapt to temporal concept drift, as it considers the predictions from different models, which might capture diverse aspects.
Ground Truth	Reframing group-robust algorithms as adaptation algorithms under concept drift, we find that Invariant Risk Minimization and Spectral Decoupling outperform sampling approaches to **class imbalance** and concept drift.

Table 3: Comparing `GPT4FS` with `GPT4FS+KG` and the original idea sentence (texts cut to fit). Overall, our analysis finds the `GPT4FS+KG` leads to suggestions with improved technical depth. However, both are far from the technical depth of innovation in original human-proposed ideas.

5 Human Evaluation & Error Analysis

We conduct human evaluations of model quality; finally, we also demonstrate the generalization of our approach to the biomedical domain.

Study I We recruit six volunteer NLP experts with graduate-level education to rate the system. Raters are told to envision an AI paper-writing assistant that suggests new ideas. We randomly select 50 instances (background+seed) from the sentence generation gold subset (§A). Each annotator is given ten instances, each paired with five system outputs including: `GPT3.5FS`, `GPT3.5Retr`, `GPT3.5FS+CT`, `GPT3.5FS+KG`, `GPT4FS`, `GPT4FS+KG`, `T5`, and `T5+SN+CL`. Raters are blind to the condition, and system outputs are randomly shuffled across instances. We observe moderately high rater agreement (Table 24 in the Appendix). We ask raters to assess idea quality by taking into account each output's relevance to the context, novelty, and whether the idea is clear and reasonable (positive ratings are dubbed "helpful" as shorthand corresponding to passing the multiple considerations). We instruct annotators to only provide positive ratings to ideas that are 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 scientific ideas from papers. Further details and evaluator guidelines are in Appendix H (sample annotations in Table 22).

Study I: Results Overall, `GPT4FS` and `GPT4FS+KG` outperform other models by a wide margin (see Table 4). Apart from GPT4, `T5+SN+CL` performs best compared to other baselines, given its stronger prior knowledge of useful background citations. In general, GPT3.5 models performed significantly 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 are longer than other baselines (Appendix D), 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 4: Percent (%) of total votes each system output receives from human raters for the idea sentence prediction task. *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 (§5).

Study II: GPT4 comparisons to real papers We conduct a follow-up human study of close competitors `GPT4FS` and `GPT4FS+KG` with a subset of the annotators. 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*.

Study II: 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.

Error Analyses 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 few-shot prompting 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..."). In general, all models tended more often than not to copy and rephrase substantial amounts of content 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". Further in-depth analyses appear in Appendices E.3,F.

Domain Generalization Case Study Our framework is domain-agnostic and can be applied to other domains by changing the IE system used in the preprocessing procedure. To demonstrate this, we conduct an additional initial experiment in the biochemical domain. We follow a similar data creation procedure as for NLP papers, outlined here briefly (full details are in Appendix I.1). 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 (Wu et al., 2023) on our data[5]. We ask two biochemical domain experts with graduate-level education to evaluate the quality of the results as before, finding them to overall rate 65% of the generated directions positively. Finally, in contrast to NLP-domain experiments, evaluators were more satisfied with the generated outputs compared to the ground truth in terms of technical detail—finding roughly 50% to be on par with the ground truth in that regard. However, this preliminary experiment was meant mainly to demonstrate the generality of our approach, and a more in-depth exploration of the utility and quality of generated outputs is left for future work.

6 Conclusions and Future Directions

We propose a new contextualized literature-based discovery task for scientific hypothesis generation, where the goal is to generate texts suggesting new ideas grounded in rich natural language context for constraining the hypothesis search space. We collect and release a benchmark dataset by mining AI-related papers, and further annotate a corpus with human evaluations to test our approaches. We present a new framework for contextualized hypothesis generation by enriching generation models to retrieve inspirations from semantic similarity graphs, knowledge graphs, and citation networks.

Our experiments demonstrate that even with advanced LLMs, the task of generating natural language scientific hypotheses is highly challenging. Ideas predicted by our approaches 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 problematic, with a huge space of potentially plausible hypotheses formulated in natural language. One interesting direction is to expand C-LBD to incorporate a multimodal analysis of formulas, tables, and figures to provide a more comprehensive and enriched background context. We will also jointly train the retrieval and generation modules to improve retrieval performance.

[5]The input and sample outputs are in Table 23.

7 Limitations

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

7.1 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 Semantic Scholar Academic Graph. We also crawled papers from PubMed 1988 to 2023/08. 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.

7.2 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/GPT4 are not finetuned and might be biased towards pretraining datasets.

7.3 Limitations of Evaluation

The automatic metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), BERTScore (Zhang* et al., 2020), BARTScore (Yuan et al., 2021), MRR, and Hits, might not be aligned with human judgment and be the best choices or our tasks. The gold subset and human evaluation are annotated by domain experts who might have biases in their selection.

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 the input context trivially leaks the output. This is also confirmed by low overall ROUGE-L scores.

7.4 Memorization Check

Carlini et al. (2023) shows 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[6], 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 series utilize most pertaining data before September 2021. Despite this, the pretraining and post-training data contain "a small amount" of more recent data.[7]
- SciBERT: Beltagy et al. (2019) states that it is pretrained on papers published before 2019.
- CS_RoBERTa: Gururangan et al. (2020) indicates that CS_RoBERTa is pretrained on papers papers before 2020.

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 can complete information such as method names or generate text that strongly mimics the ground truth papers, and found no evidence of this occurring. The PMC-LLaMA (Wu et al., 2023) is released in 2023/04 and our biochemical test set only includes PubMed papers after 2023/04.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In

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

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

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

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, 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*.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ralph A DiGiacomo, Joel M Kremer, and Dhiraj M Shah. 1989. Fish-oil dietary supplementation in patients with raynaud's phenomenon: a double-blind, controlled, prospective study. *The American journal of medicine*, 86(2):158–164.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

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. *CACM*.

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.

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.

Kanishk Jain and Vineet Gandhi. 2022. Comprehensive multi-modal interactions for referring image segmentation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3427–3435, Dublin, Ireland. Association for Computational Linguistics.

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.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and

Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. 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.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning dense representations of phrases at scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634–6647, Online. Association for Computational Linguistics.

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.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. 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.

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.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227, Online. Association for Computational Linguistics.

Tara Safavi, Doug Downey, and Tom Hope. 2022. Cascader: Cross-modal cascading for knowledge graph link prediction. In *4th Conference on Automated Knowledge Base Construction*.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-sequence knowledge graph completion and question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2814–2828, Dublin, Ireland. Association for Computational Linguistics.

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.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2017. Diverse beam search: Decoding diverse solutions from neural sequence models. In *Proceedings of the 5th International Conference on Learning Representations*.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021. Structure-augmented text representation learning for efficient knowledge graph completion. In *Proceedings of the Web Conference 2021*, WWW '21, page 1737–1748, New York, NY, USA. Association for Computing Machinery.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4281–4294, Dublin, Ireland. Association for Computational Linguistics.

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.

Qingyun Wang, Manling Li, Hou Pong Chan, Lifu Huang, Julia Hockenmaier, Girish Chowdhary, and Heng Ji. 2023. Multimedia generative script learning for task planning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 986–1008, Toronto, Canada. Association for Computational Linguistics.

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.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Towards building open-source language models for medicine. *Computation and Language Repository*, arXiv:2305.10415.

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

We download ACL Anthology papers from 1952 to 2022 using Semantic Scholar Academic Graph API[8]. We filter out papers without abstracts and not written in English to obtain 67,408 papers.

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

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 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 computer science papers. Next, we use SciCo (Cattan et al., 2021) with checkpoint from Hugging Face[9] 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)[10] 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 3 shows an example from the non-canonicalized KG corpus constructed based on the IE systems pipeline. The statistics of the non-canonicalized KG are in Table 5.

Type	#Node	#Relation
Train	70,462	61,606
Valid	11,970	8,576
Test	4,137	2,792

Table 5: Non-canonicalized KG statistics

B Training and Finetuning details

B.1 Inspiration Retrieval Module

The statistics of each neighbor type are in Table 7.

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.

[9] huggingface.co/allenai/longformer-scico

[10] github.com/allenai/sequential_sentence_classification

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 $\texttt{all}-\texttt{mpnet}-\texttt{base}-\texttt{v2}$ from Sentence-Bert (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/dual-encoder and their variants are built based on the Huggingface framework (Wolf et al., 2020)[11]. We optimize those models by AdamW (Loshchilov and Hutter, 2019) with the linear warmup scheduler[12]. Those models are finetuned on 4 NVIDIA A6000 48GB GPUs with distributed data parallel[13]. The training time for each model is about 10 hours.

B.2.1 T5 and its variants

We adopt the same input template for both tasks. For models without any neighbors, the input is the concatenation of the prompt $\mathcal{P}$ and background context $\mathcal{B}$ illustrated in §3.1 (i.e., $\mathcal{P}$ | *context:* $\mathcal{B}$). For models with neighbors, the input is $\mathcal{P}$ | *retrieve:* $n_1, n_2, .., n_k$ | *context:* $\mathcal{B}$, where $n_1, ..., n_k$ are retrieved neighbors. 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.

Idea Sentence Generation During decoding, we use beam-search to generate results with a beam size of 5 and a repetition penalty of 1.5.

Idea Node Prediction During decoding, because we want to compute HIT@10, we use beam-search to generate results with a beam size of 10 and a number of beam groups of 10. For the diversity

[11] github.com/huggingface/transformers

[12] huggingface.co/docs/transformers/main_classes/optimizer_schedules#transformers.get_linear_schedule_with_warmup

[13] pytorch.org/tutorials/intermediate/ddp_tutorial.html

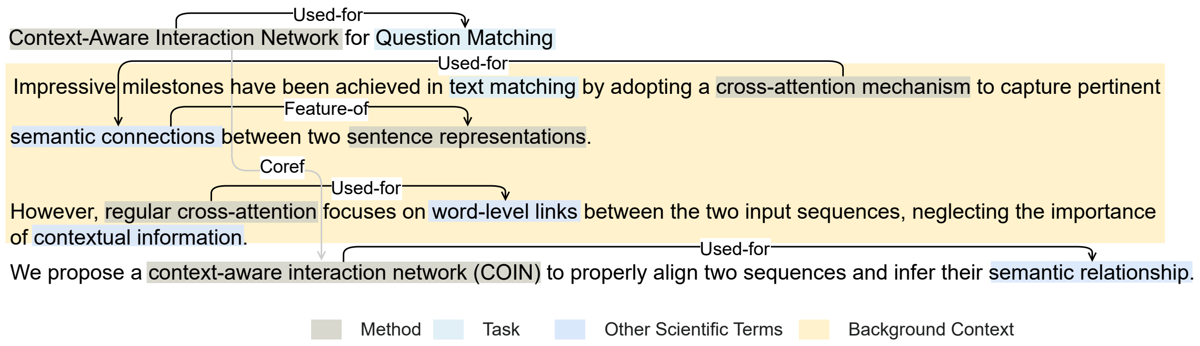

Figure 3: Preprocessing result for Hu et al. (2021) in non-canonicalized KG Corpus

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

beam search, we set the diversity penalty to 15. For the constraint beam search, we constrain the output vocabulary to all entities available in the entire KG (i.e., the KG built on all papers in our corpus).

B.2.2 GPT and its variants

We choose GPT3.5 `davinci-003`[14] (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.

Idea Sentence Generation For each selected example from the training set with *forward* relation, the template is *"Consider the following context: $\mathcal{B}$ In that context, which p can be used for v, and why? s"*, where $\mathcal{B}$ is the background context, p is the target node type, v is the seed term, and s is the target idea sentence; for *backward* relation, the template is *"Consider the following context: $\mathcal{B}$ In that context, which p do we use v, and why? s"*. For selected examples with additional retrieval neighbors, we concatenate the following additional template to the $\mathcal{B}$: *"The retrieval results are: $n_1, n_2, .., n_k$"*, where $n_1, ..., n_k$ are retrieved neighbors. For the final prompt, the template is similar to the above example template. However, the target idea sentence s will not be included. We ask the model to generate 10 outputs. We will select the best output and skip the empty output.

Idea Node Prediction The template begins with a starting prompt. For the *forward* target relation, the template is *"Suggest a p that can be used for a natural language processing v"*; for the *backward* target relation, the template is *"Suggest a p that for a natural language processing v"*. For each selected example from the training set with *forward* relation, the template is *"Consider the following context: $\mathcal{B}$ In that context, which p can be used for v? u"*, where u is the target node; for *backward* relation, the template is *"Consider the following context: $\mathcal{B}$ In that context, which p do we use v? u"*. For selected examples with additional retrieval neighbors, we also concatenate the following additional template to the $\mathcal{B}$: *"The retrieval results are: $n_1, n_2, .., n_k$"*, where $n_1, ..., n_k$ are retrieved neighbors. For the final prompt, the template is similar to the above example template. However,

[14] openai.com/api/

the target node u will not be included. We ask the model to generate 15 outputs. We will select the top 10 outputs and skip the empty output.

B.2.3 Dual-Encoder and its variants

Our dual-encoder models follow the backbone of SimKGC (Wang et al., 2022) without graph-based re-ranking. The input template is similar to T5: (i.e., $\mathcal{P}$ $\mathcal{B}$). For models with neighbors, the input is $\mathcal{P}$ | *retrieve:* $n_1, n_2, .., n_k$ $\mathcal{B}$, where $n_1, ..., n_k$ are retrieved neighbors. Our model is based on SimKGC (Wang et al., 2022)[15]. We use SciBERT [16]/CS_RoBERTa[17] as base models. We fietune those models with a learning rate of 2.5×10^{-6} and $\epsilon = 1 \times 10^{-6}$. The batch size is 150 for each GPU. The maximum training epoch for all models is 100 with 4 patience. We set the additive margin to 0.02 and 2 pre-batches for negatives. All the models are trained in mixed precision. The input length is limited to 512 tokens. Using a SciBERT/CS_RoBERTa encoder (referred to as `SciBERT` and `CSRoBERTa`) followed by mean pooling with normalization, we encode the input seed entity v and its corresponding background context $\mathcal{B}$ to get a contextualized representation $\mathbf{h}_v$. Similarly, for a potential neighbor node $u \in \mathcal{G}$, we use a second SciBERT/CS_RoBERTa encoder combined with mean pooling and normalization to get a representation $\mathbf{h}_u$. The cosine similarity between the two normalized representations is therefore $\cos(\mathbf{h}_v, \mathbf{h}_u) = \mathbf{h}_v \cdot \mathbf{h}_u$. We choose the same three types of negatives including In-batch Negatives(IB) (Karpukhin et al., 2020), Pre-batch Negatives(PB) (Lee et al., 2021), and Self-Negatives(SN) (Wang et al., 2022). During training, we use InfoNCE loss (Oord et al., 2018) with additive margin γ:

$$\mathcal{L}_{\text{cl}} = \frac{e^{((\phi(u,v)-\gamma)/\tau)}}{e^{((\phi(u,v)-\gamma)/\tau)} + \sum_{v_k \in \mathcal{N}} e^{((\phi(u,v_k))/\tau)}}$$

where $\mathcal{N}$ is the negatives, τ is the temperature, and $\phi(u, v) = \cos(\mathbf{h}_v, \mathbf{h}_u)$.

B.3 In-context Contrastive Augmentation

B.3.1 T5 and its variants

Idea Sentence Generation 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 ."*.

Idea Node Prediction For this task, we randomly select 10 scientific terms from scientific terms that appeared in the background context as in-context negatives. For example, in Figure 1, we treat scientific terms in the background context such as *"pre-training"*, *"pre-training"*, *"PLMs"*, etc.

B.3.2 Dual-Encoder and its variants

Similar to `T5`, the in-context negatives are scientific terms in the background context. We randomly select 5 scientific terms that appeared in the background context as in-context negatives.

C Evaluation Metrics

We use the official implementation of BARTScore (Yuan et al., 2021) and its corresponding checkpoint [18] for both tasks. 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 automatic evaluation on idea sentence generation for the full dataset without any filters is in Table 9

D Length for idea-sentence generation

We computed the length of the idea-sentence generation for each baseline within the human evaluation set, with results provided in Table 10. We find that the results from GPT-4 are more than twice as long as those from other baselines, suggesting GPT4 outputs contain more information.

E Automated evaluation for idea-node prediction

E.1 Evaluation Metrics

Evaluating new idea predictions given an existing scientific KG is non-trivial for non-canonicalized KGs such as ours. Traditional metrics such as MRR and Hits, which can be used for the link prediction task in canonicalized KGs, cannot deal with fine-grained/complex ground truth due to IE errors. For example, because coreference resolution cannot correctly capture all coreference relations, models might predict a correct mention that is marked

[15] github.com/intfloat/SimKGC
[16] github.com/allenai/scibert
[17] huggingface.co/allenai/cs_roberta_base
[18] github.com/neulab/BARTScore

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
Idea-Node Ground Truth	low-resource ner
Idea-Sentence Ground Truth	ELM: Data Augmentation with Masked Entity Language Modeling for Low-Resource NER

Table 8: Example (from (Zhou et al., 2022)) of retrieved neighbors for idea-sentence generation and idea-node generation. Neighbors similar to ground truth are underlined.

Model	R-L↑	BART↑	BERT↑
`GPT3.5ZS`	0.141	-6.123	0.630
`GPT3.5FS`	0.200	-5.690	0.651
`GPT3.5Retr`	0.202	-5.716	0.654
`T5`	0.260	-5.704	0.690†
`GPT4ZS`	0.120	-5.829	0.585
`GPT4FS`	0.147	-5.785	0.626
`GPT4FS+KG`	0.147	-5.792	0.626
`GPT3.5FS+SN`	0.196	-5.709	0.651
`GPT3.5FS+KG`	0.200	-5.685	0.651
`GPT3.5FS+CT`	0.184	-5.748	0.652
`T5+CL`	0.262	-5.662	0.689
`T5+SN+CL`	0.264	-5.646	0.690†
`T5+KG+CL`	0.258	-5.700	0.687
`T5+CT+CL`	**0.267***	**-5.620***	**0.690†**

Table 9: Automatic evaluation results on idea sentence generation (full dataset). *BART* denotes BARTscore. * denotes that the observed differences are statistically significant ($p \leq 0.05$).

	`3FS`	`3Rt`	`3FS+CT`	`3FS+KG`	`4`	`4+KG`	`T5`	`T5+SN`
Len	24.3	24.1	23.1	29.7	58.9	58.9	19.6	22.5

Table 10: Average length of each system output receives for human evaluation set. "`Len`" denotes the average length. "`3FS`" refers to the `GPT3.5FS`. "`3Rt`" refers to the `GPT3.5Retr`. "`4`" refers to the `GPT4FS`. "`T5+SN`" refers to the `T5+SN+CL`.

Model	MRR↑	HIT@1↑	HIT@5↑	HIT@10↑
`GPT3.5FS`	0.007	0.004	0.010	0.014
`GPT3.5Retr`	0.003	0.000	0.000	0.027
`SciBERT`	0.086†	**0.051**	0.117†	0.152
`CSRoBERTa`	0.078	0.045†	0.107	0.136
`T5+Div`	0.027	0.018	0.038	0.051
`SciBERT+CL`	0.083	0.048	0.117†	0.151
`CSRoBERTa+CL`	0.078	0.045†	0.108	0.138
`T5+Div+CL`	0.029	0.019	0.041	0.054
`GPT3.5FS+SN`	0.007	0.004	0.011	0.013
`SciBERT+SN`	0.080	0.046	0.110	0.143
`CSRoBERTa+SN`	0.073	0.041	0.099	0.132
`T5+Div+SN`	0.029	0.019	0.042	0.052
`GPT3.5FS+KG`	0.007	0.004	0.011	0.013
`SciBERT+KG`	0.079	0.044	0.111	0.144
`CSRoBERTa+KG`	0.071	0.040	0.101	0.130
`T5+Div+KG`	0.029	0.020	0.041	0.053
`GPT3.5FS+CT`	0.007	0.003	0.013	0.017
`SciBERT+CT`	**0.090†**	0.050†	**0.124†**	**0.165***
`CSRoBERTa+CT`	0.080	0.043	0.111	0.151
`T5+Div+CT`	0.045	0.032	0.065	0.076

Table 11: Canonicalized KG metrics for idea node prediction on the whole dataset.

wrong because its coreference with the target entity is undetected. Moreover, given the vastness of the search space for new ideas, numerous plausible ideas may fail to align with the ground truth precisely. Therefore, we introduce two new similarity-based evaluation metrics for natural language generation, which will compute the overlap between ground truth and predicted entities as "*soft*" Hits. Given the top-10 predictions p_i, the corresponding reference r, and one of the similarity-based metrics such as BERTScore (Zhang* et al., 2020), and BARTScore (Yuan et al., 2021), we can calculate two versions of the metric:

$$\text{AvgM} = \frac{\sum_{i=1}^{10} \frac{\text{metric}(p_i, r)}{i}}{\sum_{i=1}^{10} \frac{1}{i}} \quad (2)$$

$$\text{MaxM} = \max_{i=1,2,\ldots,10} (\text{metric}(p_i, r)) \quad (3)$$

Unlike MRR and Hits, these metrics can capture predictions that are semantically related to the ground truth, even if they don't match on the surface. Because the domain of our link prediction results is computer science, we use SciBERT (Beltagy et al., 2019) in BERTScore. Additional details of the evaluation setup are in Appendix C.

The canonicalized evaluation results are shown in Table 11. In Table 13, for each model variant we test, we compare the best dual-encoder with its T5 counterpart.[19] However, low-quality retrieval neighbors hurt model performance for baselines with relatively high performance, such as dual-encoder models. These observations echo the observations made for the sentence generation task. Different from the previous task, we observe that the in-context contrastive objective has little impact on the dual-encoder models, despite the fact that it shows consistent improvement for T5. The main reason is that the dual-encoder baselines already utilize contrastive learning objectives, while adding additional negatives might not help the model.

Type	MRR↑	HIT@1↑	HIT@5↑	HIT@10↑
Forward	**0.126**	**0.073**	**0.177**	**0.232**
Backward	0.054	0.028	0.072	0.100

Table 12: Canonicalized KG metrics evaluation results on idea node prediction for the whole dataset with SciBERT+CT.

Model	Avg BART↑	Max BART↑	Avg BERT↑	Max BERT↑
SciBERT	-6.188	-4.762	0.708	0.796
T5+Div	-6.268	-5.065	0.708	0.785
SciBERT+CL	-6.193	-4.768	0.708	0.796
T5+Div+CL	-6.267	-5.061	0.708	0.785
SciBERT+SN	-6.209	-4.803	0.708	0.794
T5+Div+SN	-6.261	-5.050	0.709	0.786
SciBERT+KG	-6.207	-4.823	0.708	0.794
T5+Div+KG	-6.285	-5.079	0.707	0.785
SciBERT+CT	**-6.150***	**-4.684***	0.710	**0.801***
T5+Div+CT	-6.191	-4.924	**0.713***	0.793

Table 13: Non-canonicalized KG metrics for idea node prediction on the whole dataset.

[19] GPT3.5 is not shown as it consistently underperforms.

Model	Avg BART↑	Max BART↑	Avg BERT↑	Max BERT↑
SciBERT	-6.783	-5.908	0.675	0.733
T5+Div	-6.750	-5.834	0.678	0.738
SciBERT+CL	-6.778	-5.913	0.676	0.734
T5+Div+CL	-6.681	-5.817	0.680	0.739
SciBERT+SN	-6.715	-5.856	0.676	0.733
T5+Div+SN	-6.752	-5.856	0.678	0.737
SciBERT+KG	-6.764	-5.909	0.675	0.733
T5+Div+KG	-6.750	-5.884	0.676	0.738
SciBERT+CT	-6.715	-5.820	0.679	0.737
T5+Div+CT	**-6.680**	**-5.797**	**0.682**	**0.739**

Table 14: Non-canonicalized KG metrics evaluation results on idea node prediction for the gold subset.

E.2 Gold evaluation

We annotate a gold dataset similar to §4.2. Because our idea node prediction task heavily relies on IE quality, we add another condition to the existing three criteria: whether the IE is of sufficient quality. The annotation details are in Appendix G. We obtain 107 pairs. The results are in Table 14. Because we only keep instances where the background context and ground truth are not similar, the scores are worse than Table 13. Furthermore, given that our models achieve a low number of correct predictions in this gold subset due to the quality of the information extraction, we decide not to use canonicalized KG metrics. Unfortunately, we find that the results do not show statistical significance between any pairs of outputs.

Model	MRR↑	HIT@1↑	HIT@3↑
SciBERT	0.671	0.434	0.933
T5+CBS	0.522	0.238	0.795
SciBERT+CL	0.672	0.434	0.934
T5+CBS+CL	0.539	0.261	0.809
SciBERT+SN	0.670	0.431	0.933
T5+CBS+SN	0.523	0.240	0.793
SciBERT+KG	0.668	0.429	0.931
T5+CBS+KG	0.518	0.231	0.793
SciBERT+CT	**0.678***	**0.444***	**0.937***
T5+CBS+CT	0.532	0.256	0.800

Table 15: Multi-choice evaluation results on idea node prediction for the whole dataset.

E.3 Multi-choice evaluation

Furthermore, we conduct a multi-choice evaluation for idea-node prediction to test the models' ability to distinguish similar candidates. Specifically, for each instance in the test set, we select three nodes that closely resemble the ground truth from the same paper. The selection is based on cosine similarity provided by SentenceBert. We ask the models to distinguish the best candidate out of four candidates, including the ground truth. Because of the constrained output space, we employ `T5+CBS` for this experiment as the sequence-to-sequence baseline. The sample input and candidates for multi-choice evaluation are in Table 17. The results are in Table 15. The results align closely with Table 11 and 13, suggesting citation neighbors successfully help models generate better results. Moreover, dual encoders are more effective when distinguishing similar candidates.

Moreover, we annotate a subset from the gold subset where all coreference and valid candidates, except the ground truth, are excluded. The results are in Table 16. However, similar to Table 14, the results between different SciBERT models are not statistically significant due to sample size.

Model	MRR↑	HIT@1↑	HIT@3↑
SciBERT	0.756	0.573	0.961
T5+CBS	0.558	0.301	0.845
SciBERT+CL	0.756	0.573	0.961
T5+CBS+CL	0.538	0.262	0.816
SciBERT+SN	**0.761**	**0.583**	0.961
T5+CBS+SN	0.554	0.301	0.835
SciBERT+KG	0.748	0.563	**0.971**
T5+CBS+KG	0.548	0.272	0.816
SciBERT+CT	0.740	0.553	0.942
T5+CBS+CT	0.568	0.320	0.806

Table 16: Multi-choice evaluation results on idea node prediction for the filtered gold dataset.

F Further result analysis

Impact of Retrieved Neighbor Similarity We also analyze the quality of different types of retrieved neighbors. We first report the number of instances with corresponding neighbors for each dataset in Table 18. We observe that KG provides the least neighbors while SN provides the most neighbors. Because our test set is built from new papers not observed during training, and our non-canonicalized KG is sparse, many nodes might not have any neighbors. Among those instances where all three neighbor types exist, we compute the semantic similarity between retrieved neighbors and the ground truth. The results are in Figure 6. For node prediction, it is evident that the citation neighbors provide the context most relevant to the ground truth, explaining the superior performance of models with citation-based retrieval in Table 11. For sentence generation, we observe that the semantic neighbors are closer to the ground truth, which matches our findings in Table 2. We note that the cosine similarities are not directly comparable across the two settings: the ground truth for node prediction is made of short entity mentions, while the ground truth for sentence generation consists of full sentences. Table 8 shows examples of retrieved neighbors for idea-sentence generation and idea-node prediction. Semantic neighbors tend to be shorter than citation neighbors, making them more similar to the idea-sentence ground truth.

Because the ground truth for sentence generation consists of full sentences, the cosine similarity is naturally lower for citation neighbors than for semantic neighbors since citation neighbors are longer than semantic neighbors (as shown in Table 19); for the same reason, however, they provide more background knowledge and thus still perform the best in Table 9.

Impact of Predicting Methods vs. Tasks As a reminder, as part of our construction, we extract salient seed terms from target sentences (§2). The seeds are mentions of scientific concepts that are linked via a "used-for" relation, resulting in tuples of the form `[head,used-for,?]` (forward direction) or `[,used-for,tail]` (backward). The head/tail mentions can loosely be considered as methods/tasks; for example, a forward prediction tuple would be `[Contrastive Learning Approach,used-for,?]`. In the node prediction setting, we aim to predict the missing tail/head. In sentence generation, we generate a full sentence that should include a reference to the missing term.

We explore differences in performance between the two directions. In Table 20, we report the scores for the forward and backward directions, using the best models in the sentence-level settings (see Table 12 for node-level). Forward predictions are better than backward predictions, even though we have more backward training pairs (Table 1). In other words, predicting tasks given methods, is easier than predicting methods given tasks.

One reason for this difference is that background context sentences naturally tend to discuss main problems in specific areas—that is, they discuss *tasks* that need to be addressed. This type of background information helps inform predictions of the missing tasks in forward-direction tuples `[method,used-for,?]`. In contrast, backward-direction tuples `[?,used-for,task]` benefit less from the background, since ideas proposed in pa-

input	context	entity	output	relation	rel_sent	Is the output trivially 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 propose d by the paper
extractive text summarizatio n 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 4: Gold subset annotation interface

Type	Content
Seed Term Prompt	symbolic reasoning is done by using OtherScientificTerm
Context	Tables are often created with hierarchies, but existing works on table reasoning mainly focus on flat tables and neglect hierarchical tables. Hierarchical tables challenge numerical reasoning by complex hierarchical indexing, as well as implicit relationships of calculation and semantics.
Ground Truth	hierarchy-aware logical form
Distractor	hierarchical structure, hierarchical tables, symbolic reasoning

Table 17: Input and candidates for multi-choice evaluation on idea node prediction

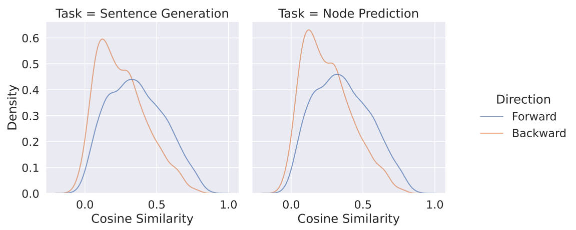

Figure 5: Cosine similarity between background context and the target node, for forward/backward edge directions. Backward edges of the form `[,used-for,task]` benefit less from background context.

Task	SN	KG	CT
Sentence Generation	69	4,122	363
Node Prediction	71	4,306	377

Table 18: Number of instances that do not have any neighbors for the test set.

Task	SN	KG	CT
Sentence Generation	3.11	4.09	8.73
Node Prediction	3.12	4.10	8.74

Table 19: Average number of tokens for retrieved neighbors and the target where all three neighbors exist

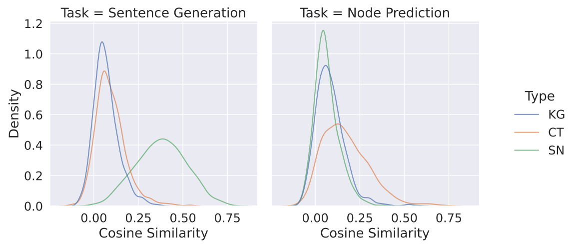

Figure 6: Cosine similarity between the ground truth and the least similar retrieved neighbor, where all three neighbors exist.

Type	R-L↑	BART↑	BERT↑
`Forward`	**0.287**	**-5.557**	**0.698**
`Backward`	0.247	-5.681	0.683

Table 20: Automatic evaluation results on idea sentence generation for the whole dataset with `T5+CT+CL`.

pers often suggest new *methods* to address a given background task. See also Figure 5.

Impact of Retrieval and In-Context Contrastive Objective We conclude with qualitative observations on the impact of retrieval and our in-context contrastive objective. We observe that the retrieved neighbors enrich the input context with relevant knowledge. For example, given the seed term *"multi-modal interactions"* as a method, and the cited paper title *"cross-modal relationship inference for grounding referring expressions"*, `SciBERT+CT` correctly predicts *"referring image segmentation"* as the second-highest ranked task (relation extracted from Jain and Gandhi (2022)).

We also confirm qualitatively that the in-context contrastive objective reduces the model's tendency to copy directly from context. For example, given the seed term *"graph neural networks (gnns)"* and context *"...interest in word alignments is increasing again for their usefulness in domains such as typological research..."*, `SciBERT` directly copies "alignments" as part of the top result, while `SciBERT+CL` predicts *"bilingual graphs"* which is closer to the ground truth *"multi parallel word alignment"*.

G Gold Dataset Annotation Details

The gold dataset annotation interface is in Figure 4. 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.

H Human Evaluation Details for Idea-Sentence Generation

The instructions for human evaluation can be found in Figure 7, while an example of the human evaluation interface is provided in Figure 8 and 9. 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 21 and the human labels are in Table 22. The human annotation agreement is in Table 24.

I Biochemical Case Study

I.1 Dataset Collection

We collect PubMed papers from 1988 to 2023 using Entrez Programming Utilities API[20]. 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. Similar to the construction of the silver dataset, we only keep samples with low similarity between background context and corresponding ground truth sentences [21]. Our final dataset has 6,491 papers before 2023/02, 778 papers from 2023/02 to 2023/04, and 663 papers after 2023/05.

I.2 Training Details

Our PMC-LLaMA and its variants are built based on the Huggingface framework (Wolf et al., 2020)[22]. We use its `chaoyi-wu/PMC_LLAMA_7B_10_epoch` as the base model. We finetune those models with a learning rate of 6×10^{-6} and $\epsilon = 5 \times 10^{-7}$. The maximum training epoch for all models is 15. All models are fine-tuned on 4 NVIDIA A100 80 GB GPUs with Fully Sharded Data Parallel[23]. The training time for each model is about 20 hours. The input and sample outputs are in Table 23.

J Scientific Artifacts

We list the licenses of the scientific artifacts used in this paper: Semantic Scholar Academic Graph

[20] `www.ncbi.nlm.nih.gov/books/NBK25501/`
[21] The similarity is calculated with `all-mpnet-base-v2`.
[22] `github.com/huggingface/transformers`
[23] `https://huggingface.co/docs/accelerate/usage_guides/fsdp`

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 21: 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 22: 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	AbA
Context	Complex sphingolipids and sterols are coordinately involved in various cellular functions, e.g. the formation of lipid microdomains.
PMC-LLaMA	AbaA domains could be classified into different groups, including GrA, GrB, GrC, GrF, and AbaA domains.
PMC-LLaMA + CT	Among them, the clinical severity is an important question. Furthermore, we would like to explore the effects of **AbA** on the development of the severity, onset of biogenic factors and the severity/pharmacokinetic drug interaction.
PMC-LLaMA + KG	Furthermore, we found that **AbA** utilization was significantly reduced in the absence of Bs1.
PMC-LLaMA + SN	When A/N inoculum size) the most abundant PAH on LF is significantly lower than that of Arab1.

Table 23: Input and idea-sentence generation results for seed chemical "***AbA***" 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 24: Percent (%) of same labels from overlapped 10 human evaluation instances on each pair of annotators.

API (API license agreement[24]), Huggingface Transformers (Apache License 2.0), SBERT (Apache-2.0 license), BARTScore (Apache-2.0 license), BERTScore (MIT license), SimKGC (no license), PMC-LLaMA (no license), Entrez Programming Utilities API (Copyright[25]), PubTator 3 (Data use policy[26]), and OpenAI (Terms of use[27]).

K Ethical Consideration

The contextualized literature-based discovery task and corresponding models we have designed in this paper are limited to the natural language process-

[24] api.semanticscholar.org/license/

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

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

[27] openai.com/policies/terms-of-use

ing (NLP) domain, and might not apply to other scenarios.

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

K.2 Data Collection

We collect 67,408 ACL Anthology papers from 1952 to 2022 using Semantic Scholar Academic Graph API, under API license agreement[28]. 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 7,932 PubMed papers from 1988 to 2023 using Entrez Programming Utilities API[29]. We follow their data usage guidelines[30].

[28] `https://api.semanticscholar.org/license/`
[29] `www.ncbi.nlm.nih.gov/books/NBK25501/`
[30] `www.ncbi.nlm.nih.gov/books/about/copyright/`

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 7: 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 8: 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 9: Human evaluation example for GPT3.5Rnd+KG, GPT4Rnd, and GPT4Rnd+KG