LinkNav: Surfacing Interconnected Information in Scientific Articles

Sebastian Joseph[1] **Jennifer Healey**[2] **Junyi Jessy Li**[1] **Ani Nenkova**[2]
[1]The University of Texas at Austin, [2]Adobe Research
{sebaj,jessy}@utexas.edu, {jehealey,nenkova}@adobe.com

Abstract

We present LinkNav[1], an enhanced experience for reading academic papers which makes explicit connections between related but non-adjacent passages. To create the experience, we instruct a language model to generate questions that may arise while reading a passage and then search for answer passages elsewhere in the document, forming intra-document connections when answers are found. We confirm that these building blocks work well to power the experience, with an answer detection pipeline that works with high precision, resulting in a reasonable number of connections being made for a document. On a dataset of academic papers, we find that connected passages are on average ten segments away from each other, making explicit connections that a reader may have otherwise missed.

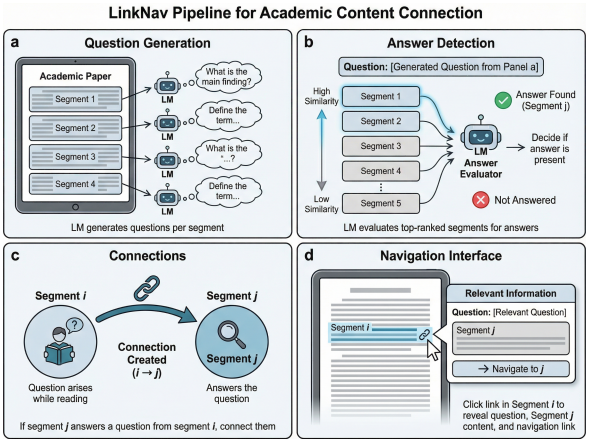

Figure 1: LinkNav pipeline and experience. *(a)* A paper is broken down into segments and a language model generates questions for each segment i. *(b)* Similarity is computed between the questions and all other segments. An LLM decides if one of the top five segments, j is the answer. *(c)* If so, i and j are considered connected. *(d)* Connections are surfaced via links and a side panel.

1 Introduction

Academic papers present hypotheses, experiments, mathematical derivations and evaluative statements about a complex body of work that have to be presented in linear form. In this complex landscape, it is normal to have semantic connections between segments of writing that are not adjacent in their linear presentation and that may be missed even by an attentive reader. A non-linear reading order of academic literature is common (Lo et al., 2023) with readers at different career stages approaching it differently (Hubbard and Dunbar, 2017).

The need to connect information is recognized by authors, who make explicit connections via pointers to sections and appendices. It also surfaces during informal academic talks, where it is commonplace for the audience to ask a question and the presenter to respond that the answer is to be discussed at a later point in the presentation. In this paper we present analyses of questions drawn from peer reviews of scientific papers (cf. Section 3); reviewers frequently ask questions that are already answered in a paper, further supporting the need and potential value of surfacing the connection between non-adjacent content.

Figure 1[2] schematically introduces LinkNav, an augmented reading interface which supplies connections even if the author did not make them explicit and displays the relevant content. Connections are established via the generation of inquisitive questions, an approach shown to relate to reader expectations (Wu et al., 2024). LinkNav also provides a link to navigate to the related segment in the paper. Future user studies will show if people prefer the localization of information or navigation in the linear paper presentation; LinkNav supports both. Figure 2 shows a screenshot of the experience reading a paper in the interface, along with one of the questions and the extractive (author wording

[1]LinkNav link: linknav.netlify.app

[2]Rendered with Gemini following a detailed author prompt.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 453–462

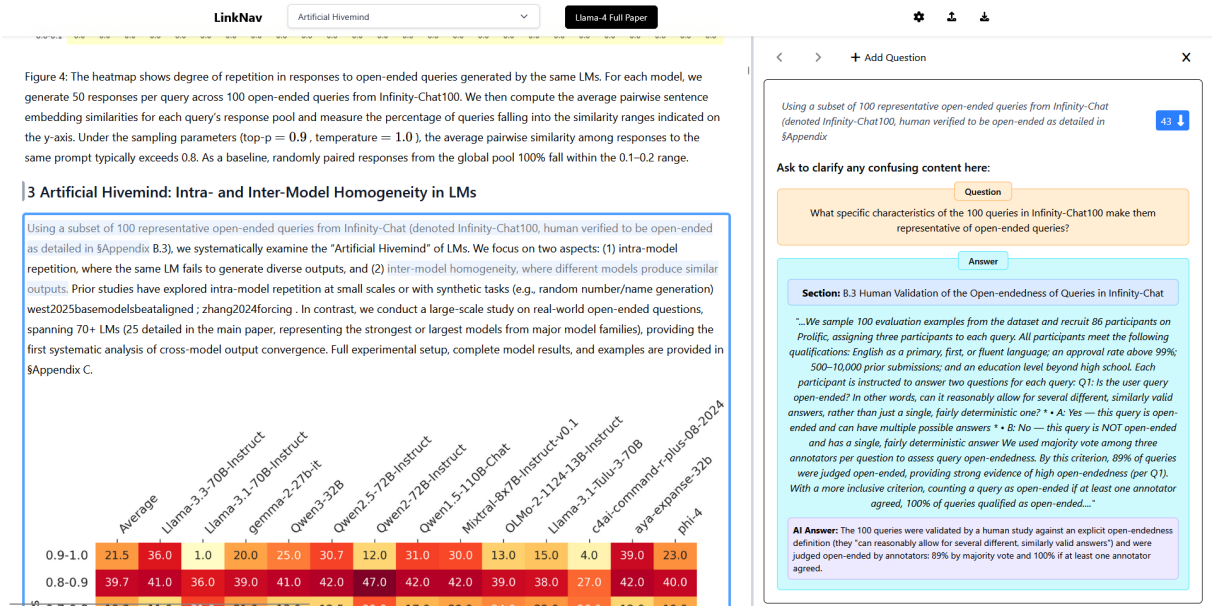

Figure 2: LinkNav surfaces links between different segments in papers. Text portions that likely triggered the question are displayed as links. Clicking on the link reveals a panel that displays the question, an extractive answer from the paper, a generative answer by a model presented with the question and the segment containing the answer. A link allows the reader to navigate to the segment where the answer is located. The distance between the originating segment and the answer segment is displayed. A quick glance will let the reader decide if they want to read the linked segment now, perhaps if it is very far away, or continue reading if the linked segment is nearby.

answer) and a generative answer (from a language model) provided with the question and the paper.

Using a subset of the PeerQA dataset (Baumgärtner et al., 2025), we analyze the methodologies used for this experience and the connections they create. Our analyses show that LinkNav reveals explicit connections between distant areas of a document that a reader could have likely missed. Links connect for example information in appendices to the place where content is discussed, evaluative statements in the abstracts and the introduction to segments in the results and experiments sections of papers, giving the reader the ability to either skim more efficiently or do a close read with relevant details from the entire paper available where the reader is currently paying attention.

2 Related Work

LinkNav is *an augmented reading* experience driven by *question generation*. We highlight the key connections with these lively research areas.

Augmented reading Most closely related to our work is the Qlarify augmented reading experience (Fok et al., 2024) for paper abstracts. There some entities are highlighted to indicate that they can be expanded. One contextual LLM-generated question about the entity is revealed when hovering over the link, while three static questions are always present for each entity—*Define*, *Expand*, *Why*—which provide definition, concrete details about a concept and explanations about motivation. Readers can select text in the abstract and ask their own questions. The answers are linked to paper content supporting the answer, allowing readers to navigate to that part of the paper on demand.

A user study with doctoral students revealed that readers appreciated the cognitive load reduction provided by the suggested question. They asked their own questions only a quarter of the time. A field study in which Qlarify was provided to conference attendees to browse the newly available proceedings confirmed the preference for pre-generated questions. In that setting, almost 90% of interactions were via the three static questions. In a comparison with an interface that only allowed readers to ask their own questions instead of offering LLM generated questions, readers asked half as many questions. These findings inform our approach, emphasizing the importance of pre-surfacing interlinked content.

ScholarPhi (Head et al., 2021) is an augmented reading experience for full academic papers. It solves only for the need for just-in-time definitions of mathematical symbols used in equations and

derivations. Like our work, it caters to a need occurring during reading of the full paper by bringing together information that is needed for understanding but is not adjacent in the linear presentation. DocVoyager (Lee et al., 2025) on the other hand expands the Qlarify idea of recursive invocation of information driven by questions, offering readers tailored non-linear reading experience. In contrast, LinkNav preserves the presentation flow of the author and leaves full control to the reader when they want to check the connection, without presupposing that it will guide the reader in their journey through the document. Information cards for referenced papers, citing sentences from recently read papers, explanations and synthesis are also helpful to readers (Lo et al., 2024).

Question Generation Suggested questions (Huang et al., 2023; Wang et al., 2019; Lin et al., 2024) are now common in AI reading assistants. These are questions whose answers reveal noteworthy content in the document and that are made available to the reader to initiate and continue engagement with the assistant; they do not expose interconnected content.

Similarly for medical documents (August et al., 2023), "key questions" were used to guide readers to the answer, resulting in a non-linear reading order that is more efficient without loss in comprehension. These findings related to key questions demonstrate the potential of a question-generation approach. In contrast to our work, key questions are akin to a fixed set of FAQs from clinicians, rather than generated dynamically based on context.

Questions most similar in function to the ones that underpin LinkNav are inquisitive questions, which may be evoked in the mind of a reader during comprehension (Ko et al., 2020; Westera et al., 2020; Wu et al., 2024). Some of the inquisitive questions prepare the reader for content that will be presented later in the article. These questions typically have high *salience* and readers have high expectation that they will be answered. (Wu et al., 2024). Other questions may not be answered because of gaps in cultural, topical or other background knowledge by the reader while the author assumes them as common ground.

Earlier work on annotation of sentence specificity in news articles has revealed that different people find similar parts of the sentence to be underspecified (Li et al., 2016). This prior work, along with Westera et al. (2020) and Ko et al. (2022), also found that often questions evoked in a sentence are answered in the immediately following sentence, as authors likely anticipate the question the reader may have or explicitly have phrased their content to make a piece of information salient in the reader's mind. A fair amount of questions were found to be answered *before* the place where the question arose, with writing organized in a way that minimizes missing information. We incorporate these findings into LinkNav by pruning these author crafted links to content prior to a segment, or to the same or immediately following segment. We enrich the reading only with segments that occur after the place where the question arises.

Generating questions that probe gaps in the knowledge conveyed by two texts or possessed by different people is useful in many scenarios such as measuring information loss during text simplification (Trienes et al., 2024) or text editing (Cole et al., 2023), education (Rabin et al., 2023) and automated paper reviewing (Chang et al., 2025). These scenarios are unrelated to augmented reading even though question generation is central ingredient in the solutions they provide.

3 Reviewers May Benefit From LinkNav

Data We use the PeerQA dataset (Baumgärtner et al., 2025) for formative analyses here and for evaluation of LinkNav components later. PeerQA consists of questions contained in peer reviews of (mostly) machine learning and natural language processing venues. For part of the PeerQA corpus, the authors of the published paper annotated—in the camera ready version of the paper—the spans that answer the review question or marked the question as unanswerable from the camera ready paper.

Analysis To show the need for augmented reading for papers, we collected from OpenReview the versions of the papers submitted for review. We were able to find the submission version for 116 of the PeerQA papers with author answers.

For each answer span annotated in the camera ready version, we search for a corresponding span in the submitted version of the paper, looking for exact text overlap. We find that 52.1% of the 261 questions marked as answered in the camera ready are also answered in the version submitted for review. The percentage of all PeerQA questions (362) from these papers that can also be answered in the submitted version is 37.6%.

LinkNav functionality would have surfaced

some of the connections that reviewers missed. Reviewing is one setting in which the reader is doing a close reading of the paper and the omissions are most likely an indication of the complexity of the academic paper genre. Explicit linking and surfacing of content can help skimming starting from the abstract or closer reading as in reviewing.

4 Putting LinkNav Together

First, an article is split into approximately 512 token segments, adjusted for paragraph and section boundaries and the presence of figures or tables. Figures and tables are considered their own paragraphs, and segments are built by continually adding paragraphs until adding the next paragraph would exceed the 512 token limit. We also ensure that at section boundaries a new segment is created to prevent text from multiple sections appearing in a single segment.

LinkNav functionality is supported by three components as shown in Figure 1: *(a)* question generation of plausible questions that may arise while reading a segment; *(b)* answer detection to find if an answer to that question exists in the paper. Detection is done by computing similarity between a question and paper segments. The top most similar segments and the question are then passed to a language model, which decides if any of these segments answer the question; *(c)* create a directed graph structure that links segments i and j if a question that is generated for segment i is answered in segment j. Questions for which an answer is not found in the paper are discarded and do not result in a link between segments. Each component can be instantiated in a more sophisticated manner, likely leading to improved performance but certainly more costly.

Here we evaluate if the solution with fairly cost-effective components can be deemed sufficient for a minimal viable experience. For evaluation we use a randomly sampled set of 29 papers from the PeerQA papers with author annotated question answers. We call this subset PeerQA-Gold because of the availability of highly accurate gold answers. This subset is large enough to provide meaningful measurements but not unreasonably costly to compute the results.

4.1 Question Generation

To generate questions, we pass each segment to `llama-4-maverick-17B-128E-Instruct` with the prompt shown in the Appendix A, to obtain meaningful questions that can arise while reading that segment. Prior work has shown that the vast majority of questions generated in this manner are valid, reasonable questions (Wu et al., 2023, 2024).

Model	#Qs	% Ans	Segment Qs	Distance
GPT-4o	4583	61.292	5.348	11.503
o3	4741	32.272	5.532	11.078
Llama-4	3558	79.174	4.152	10.491

Table 1: Number of questions generated (#Qs), percent of answerable questions (% Ans), average number of questions generated for a document segment (Segment Qs), and average distance in segments between connecting segments for valid connections (Distance) for three different question-generation models.

Table 1 shows statistics for the questions generated for the PeerQA-Gold subset papers with three models: two closed-source models, `gpt-4o` and `o3`, and one open-source model `llama-4`. For this analysis only, we use a slightly different heuristic for splitting documents into segments, only taking into account section boundaries and the 512 token limit to ensure a more uniform set of segments. The models generate a different number of questions, with `Llama-4` producing the fewest questions. However `Llama-4` generates the largest number questions that are answered in the paper; about 80% of the questions it generates are answered somewhere in the document. In contrast, `o3` generates many more questions, but only over a third of them are answered in the paper itself. Despite these differences, ultimately all three models generate five connections per segment via questions that are answerable later in the paper. The segments where the question arises and the segment that contains the answer are 10 segments apart from each other on average.

Given the similar distribution of questions per segment and the distance between connected segments for all three models, we choose to use `Llama-4`. `Llama` creates a similar number of connections as the other models but with considerably fewer questions for which answers have to be found in the document. The LinkNav experience with `Llama-4` is thus faster to create and cheaper than using the other models.

Although all three models produce a similar number of connections and comparable distances between linked segments, they often connect different segments. Table 2 shows the pairwise over-

System Pair (A, B)	A Unique	B Unique	Common
(GPT-4o, o3)	53.67	28.70	17.63
(GPT-4o, Llama-4)	41.37	34.54	24.09
(o3, Llama-4)	33.15	51.52	15.33

Table 2: For each model pair (A, B), the percentage of connections made by A only (A Unique), by B only (B Unique), and by both systems (Common) out of the union of connections, averaged per paper.

lap in connections for the PeerQA-Gold papers. Across all pairs, the majority of connections made by each system are *not* shared with the other, with unique connections consistently outweighing common ones. Evaluating human preference for specific connections is beyond the scope of this work, but these substantial differences warrant a future comparison of perceived differences in the resulting experience.

4.2 Answer Detection

After the questions are generated, answer detection finds if an answer to each question exists in the paper. To keep runtime and cost reasonable, detection is done by computing similarity between the question and paper segments. The top most similar segments and the question are then passed to a language model, which decides if any of these segments answer the question.

Specifically, we use the OpenAI `text-embedding-small` embeddings to find the n paper segments that are most similar to the question. We seek the smallest n that would ensure that the correct answer is among the top n, to minimize the length of text passed on to the model making the final decision if the question is answered, while ensuring that most questions with an answer are answered.

Figure 3 shows the cummulatative distribution function for the percent of answerable questions in the PeerQA-Gold dataset. For the first five steps adding an extra segment increases the coverage of true answers by over 5% at each steps. After that, the growth slows down, with 95% of answers in the top 30 segments. We choose to pass on the top 5 segments to the model for final decision. At this cut-off, 72% of questions with known gold annotated answer have a segment answer in the top 5. Like this, we lose some connections in the text, trading these off for speed and cost.

After that we use a language model to decide if any of the five segments answers the question. If

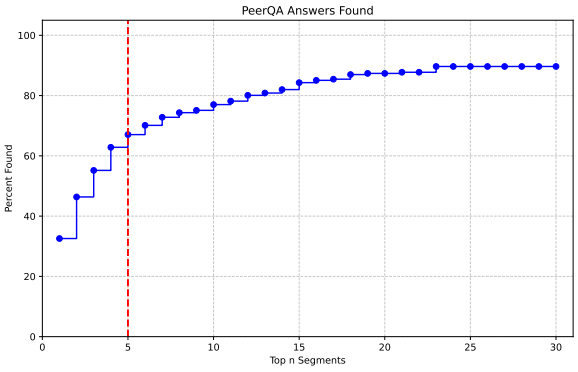

Figure 3: Cumulative percent of PeerQA-Gold answers in the Top n segments most similar to the question. Beyond the top-5 segments, the growth in this cumulative percent slows down significantly.

none of the segments is an answer to the question, the question is not suitable for content linking. If a segment is identified as answering the question, the segment where the question arose and the answer segment are connected in LinkNav.

In table 3 we present results for `gpt-5-mini` and `gpt-4o` on PeerQA-Gold for finding an answer to a question q among the five segments most similar to q. Precision indicates how often a segment identified as an answer to the question is indeed an answer and recall indicates what percentage of existing answers are identified.[3] We present the prompt in Appendix B.

Model	Precision	Recall	F1
`gpt-5-mini`	0.798	0.801	0.799
`gpt-4o`	0.836	0.586	0.689

Table 3: Precision, recall, and F1 @ 5 (among the top 5 answers) for models deciding if a segment is an answer to a given question. Evaluated against ground truth PeerQA-Gold.

The precision of 4o is about 5% higher. Precision is important for the LinkNav experience because if two segments are connected in the interface but the connected segment does not answer the question that triggered the link, the experience will not be intuitive. With 80% precision, most of the connections will be well-justified and either model can be used. For recall however, `gpt-5-mini` is 20% absolute difference better than `gpt-4o`. `gpt-4o` will miss 40% of connections

[3]For 28% of the questions the answer segment is not among the top five as seen in Figure 3, lowering the maximum obtainable recall. Some questions have more than one segment that answer the question, so recall can exceed 72%.

between segments, leading to an incomplete experience. `gpt-5-mini` is cheaper, faster and overall more suitable for use in LinkNav.

5 Properties of Connections

Model	Before	After	Same	Connect
GPT-4o	27.67	7.53	30.15	1369
o3	27.78	7.52	26.24	848
Llama-4	28.93	7.22	33.60	1146

Table 4: Percent of unfiltered question-answer connections connecting backwards (Before), percent of unfiltered question-answer connections connecting to immediately forward adjacent segments (After), percent of unfiltered connections connecting to the same segment as the question (Same), and number of valid connections after filtering (Connect).

System	$d = 0$	$d = 1$	$d = 2$	$d = 3$	$d > 3$
GPT-4o	45.040	19.837	12.369	7.468	15.286
o3	52.509	23.337	11.552	5.484	7.118
Llama-4	46.558	21.937	10.268	9.802	11.435

Table 5: Percent of all segments corresponding to their in-degree d for all three systems evaluated.

Here, we present additional statistics about the segment connections established via each of the three models, `gpt-4o`, `o3` and `llama-4`.

Answers that appear before or in near proximity to the question origin are presumably low utility, as the reader have either read or will soon read the information that will address that question. So all question-answer connections where the answer segment preceded, coincided with, or immediately followed the question's origin segment are filtered out in the current implementation. Ideally in future work instead we will have the ability to generate only valid questions that link further segments following the point of reading.

The percentage of filtered connections by type is shown in Table 4. The three models have roughly similar rate of questions leading to invalid connections between segments. About one third of all questions are answered in a segment before the segment where they arise. Presumably the authors organized their writing to have this effect, so details are already shared at the point where the reader needs them. In future work, these planned connections can also be leveraged in an augmented experience, with an info card that summarizes previously shared content. Questions however are not a good vehicle for introducing the reminder.

Another 30% of questions are answered in the segment for which they were generated. Augmented reading for these is clearly unnecessary. `o3` has the fewest (25%) of these types of questions.

Another 7% of connections are to segments immediately following the one where the question arose, again suggesting minimal utility at best for making the connection explicit in LinkNav.

Overall, about 65% of generated questions are useless, requiring generation and answer validation time and cost but not leading to a meaningful connection. More efficient methods for question generation will be valuable in making the experience practical in the future.

Question Yield The number of questions and the number of connections that could be extracted from these questions are useful factors to consider for document-linking question generation. For each question generated, there is an additional cost in detecting its answer within a document. However, the questions generated should also yield utility by revealing a connection. Thus, a higher proportions of answerable questions are desirable. These factors are what led us to choose `llama-4-maverick-17B-128E-Instruct` as the question generation model in LinkNav.

In-degree Analysis A segment's in-degree d is equal to the number of different questions answered by information in that segment. High in-degree indicates that a segment contains important information relevant to content presented throughout the paper. Pre-neural extractive summarization methods have highlighted that a text node with highest in-degree usually contains the most important information in the text (Mihalcea and Tarau, 2004; Erkan and Radev, 2004). For the present LinkNav experience however repeated connections to the same content may be distracting if the reader had already read the central segment via a previously created link.

One way to deal with repeated links is to make the text links fade in color once a segment has been visited during reading. Like this, the reader will be aware that they have already seen the information. A cruder way is to attempt to minimize links to the same segment, even if they naturally exist in a text.

In Table 5, we present the percent of segments where $d = 0, 1, 2, 3$ and $d > 3$ for each source of questions. Questions generated with o3 leave over

half of the segments in the paper without connections, an in-degree of zero. GPT-4o and Llama-4 have much smaller number of unconnected segments. Out of all evaluated models, `llama-4` resulted in the highest proportion segments with in-degrees ranging from 1 to 3. This result combined with the factors mentioned earlier motivated our decision to use it as the question generation model for LinkNav.

6 Conclusion

Academic papers encode rich semantic connections within a linear structure, often leaving readers to bridge distant passages on their own. We introduce LinkNav, which surfaces such non-adjacent connections by generating reader-aligned questions and identifying answer-bearing segments through dense retrieval and model-based verification. Our evaluation shows that many linked passages are substantially far from each other. By combining localized answers with direct navigation, LinkNav augments the traditional paper with a semantic layer that supports more coherent and efficient reading.

Limitations

Our work deals with the problem of surfacing distant semantic connections between segments of a paper. Readers however may need additional types of support, such as identifying key segments and their interpretation and reminders for content they have already read. The analysis of the paper graph shows that it may support further augmentations but these are beyond the scope of this work.

In our analysis of LLM-generated questions for intra-document connections, we used one-fourth of the papers we extracted from the PeerQA dataset. This was done mostly for cost and time reasons. The smaller dataset is sufficient to meaningfully measure common events such as number of questions generated per segment, answer recall when using question-segment similarity and precision and recall for the final answer detection.

Our question generation and subsequent answer detection approaches results in possibly unanswerable questions and low-utility connections that have to be filtered out later. The choice of the model we used for question generation was predicated on minimizing this waste. This paper is centered on this application of surfacing intra-document connections to users via inquisitive questions. Future work will focus on approaches for further optimizing question generation for such connections.

As with any user experience, there is the potential for certain subsets of readers to find low utility in this application. We highlight clear benefits for readers using LinkNav, and future work will focus on user studies to understand readers preferences for surfacing connections in academic papers.

Ethical Concerns

Large language models are transforming society, for some making human reading unnecessary. We believe that human competency even in the future will require human reading and seek ways to modernize the experience for doing so.

We use evidence from peer review that reviewers miss some connections and look for information that the authors have already provided in 30% of their questions. This analysis is not a critique of reviewers, but an evidence that reading is complex and has to be modernized with the emerging capabilities offered by large language models.

Acknowledgements

We thank Jack Wang and Alexa Siu for their valuable feedback and comments in the early stages of this work.

References

Tal August, Lucy Lu Wang, Jonathan Bragg, Marti A. Hearst, Andrew Head, and Kyle Lo. 2023. Paper plain: Making medical research papers approachable to healthcare consumers with natural language processing. *ACM Trans. Comput.-Hum. Interact.*, 30(5).

Tim Baumgärtner, Ted Briscoe, and Iryna Gurevych. 2025. Peerqa: A scientific question answering dataset from peer reviews. *Preprint*, arXiv:2502.13668.

Yuan Chang, Ziyue Li, Hengyuan Zhang, Yuanbo Kong, Yanru Wu, Zhijiang Guo, and Ngai Wong. 2025. Treereview: A dynamic tree of questions framework for deep and efficient llm-based scientific peer review. *Preprint*, arXiv:2506.07642.

Jeremy R. Cole, Palak Jain, Julian Martin Eisenschlos, Michael J.Q. Zhang, Eunsol Choi, and Bhuwan Dhingra. 2023. DiffQG: Generating questions to summarize factual changes. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3088–3101, Dubrovnik, Croatia. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479.

Raymond Fok, Joseph Chee Chang, Tal August, Amy X. Zhang, and Daniel S. Weld. 2024. Qlarify: Recursively expandable abstracts for directed information retrieval over scientific papers. *Preprint*, arXiv:2310.07581.

Andrew Head, Kyle Lo, Dongyeop Kang, Raymond Fok, Sam Skjonsberg, Daniel S. Weld, and Marti A. Hearst. 2021. Augmenting scientific papers with just-in-time, position-sensitive definitions of terms and symbols. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA. Association for Computing Machinery.

Baorong Huang, Juhua Dou, and Hai Zhao. 2023. Reading bots: The implication of deep learning on guided reading. *Front. Psychol.*, 14:980523.

Katharine E Hubbard and Sonja D Dunbar. 2017. Perceptions of scientific research literature and strategies for reading papers depend on academic career stage. *PloS one*, 12(12):e0189753.

Wei-Jen Ko, Te-yuan Chen, Yiyan Huang, Greg Durrett, and Junyi Jessy Li. 2020. Inquisitive question generation for high level text comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6544–6555, Online. Association for Computational Linguistics.

Wei-Jen Ko, Cutter Dalton, Mark Simmons, Eliza Fisher, Greg Durrett, and Junyi Jessy Li. 2022. Discourse comprehension: A question answering framework to represent sentence connections. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11752–11764, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yoonjoo Lee, Nedim Lipka, Zichao Wang, Ryan Rossi, Puneet Mathur, Tong Sun, and Alexa Siu. 2025. Docvoyager: Anticipating user's information needs and guiding document reading through question answering. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '25, New York, NY, USA. Association for Computing Machinery.

Junyi Jessy Li, Bridget O'Daniel, Yi Wu, Wenli Zhao, and Ani Nenkova. 2016. Improving the annotation of sentence specificity. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Zihao Lin, Zichao Wang, Yuanting Pan, Varun Manjunatha, Ryan Rossi, Angela Lau, Lifu Huang, and Tong Sun. 2024. Persona-sq: A personalized suggested question generation framework for real-world documents. *Preprint*, arXiv:2412.12445.

Kyle Lo, Joseph Chee Chang, Andrew Head, Jonathan Bragg, Amy X. Zhang, Cassidy Trier, Chloe Anastasiades, Tal August, Russell Authur, Danielle Bragg, Erin Bransom, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Yen-Sung Chen, Evie Yu-Yen Cheng, Yvonne Chou, Doug Downey, Rob Evans, and 36 others. 2023. The semantic reader project: Augmenting scholarly documents through ai-powered interactive reading interfaces. *Preprint*, arXiv:2303.14334.

Kyle Lo, Joseph Chee Chang, Andrew Head, Jonathan Bragg, Amy X. Zhang, Cassidy Trier, Chloe Anastasiades, Tal August, Russell Authur, Danielle Bragg, Erin Bransom, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Yen-Sung Chen, Evie Yu-Yen Cheng, Yvonne Chou, Doug Downey, Rob Evans, and 36 others. 2024. The semantic reader project. *Commun. ACM*, 67(10):50–61.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Roni Rabin, Alexandre Djerbetian, Roee Engelberg, Lidan Hackmon, Gal Elidan, Reut Tsarfaty, and Amir Globerson. 2023. Covering uncommon ground: Gap-focused question generation for answer assessment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 215–227, Toronto, Canada. Association for Computational Linguistics.

Jan Trienes, Sebastian Joseph, Jörg Schlötterer, Christin Seifert, Kyle Lo, Wei Xu, Byron C. Wallace, and Junyi Jessy Li. 2024. Infolossqa: Characterizing and recovering information loss in text simplification. *Preprint*, arXiv:2401.16475.

Siyuan Wang, Zhongyu Wei, Zhihao Fan, Yang Liu, and Xuanjing Huang. 2019. A multi-agent communication framework for question-worthy phrase extraction and question generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7168–7175.

Matthijs Westera, Laia Mayol, and Hannah Rohde. 2020. TED-Q: TED talks and the questions they evoke. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1118–1127, Marseille, France. European Language Resources Association.

Yating Wu, Ritika Mangla, Alexandros G. Dimakis, Greg Durrett, and Junyi Jessy Li. 2024. Which questions should I answer? salience prediction of inquisitive questions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19969–19987, Miami, Florida, USA. Association for Computational Linguistics.

Yating Wu, Ritika Mangla, Greg Durrett, and Junyi Jessy Li. 2023. QUDeval: The evaluation of questions under discussion discourse parsing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5344–5363, Singapore. Association for Computational Linguistics.

A Question Generation Prompt

Prompt A

```
System: You are logical, intelligent, insightful, precise, and can understand the contents of research papers. You are
knowledgable on different fields and domains of science and engineering. You are able to interpret research papers,
create questions and answers, and compare multiple aspects.

Imagine the following scenario: You are reviewing a submitted research paper manuscript. After you are done reading, you
have questions you want to ask the authors. You are a very intelligent reader so you don't ask question that are already
answered within the paper.

I will provide you two things: First, context which you have already read, and a chunk of text that you are currently
reading.

This is your task. First, you need to look at the context that is provided to you. This is information that you have
already read and aware of. Please don't ask questions that can be answered through this context. What you need to do
instead is generate all possible follow-up questions arising from the provided chunk of text that you are currently
reading. These follow-up questions must be insightful and highlight a critical gap of information that a reader would
desire to know while reading.

You MUST follow these rules when creating these questions:
(1) The question must not be answered by text in the context. Make sure to thoroughly read the text in the context and
craft a question that cannot be answered by it.
(2) These questions must be questions that a smart reader is naturally thinking about when reading this chunk of text.
Remember, they also have already read the context as well, so they will not be thinking of questions that they already
know the answer to. Put yourself in this smart reader's shoes when crafting these questions.

I will provide you with a typology of these questions below. You can use this typology to come up with more diverse
questions.

Typology:
  - Content:
    - content-clarification: questions that try to clarify something that you've read
  - Insights:
      - insights-more: questions that ask for more insights
      - insights-nuance: questions that ask for more nuance in the insights
      - insights-soundness: questions that ask whether the insight is sound.
  - Measurement:
      - measurement-more: questions that ask for more metrics to evaluate in experiments
      - measurement-detail: questions that ask for more detail about some measurement/metrics
      - measurement-alternative: questions that ask why an alternative measurement wasn't used
  - Method:
      - method-alternative: questions that probe at an alternative method that has a similar effect
      - method-detail: questions that probe for more detail about the method
      - method-motivation: questions that probe at the motivation for using such a method

When you respond, first think really hard, step-by-step, about every span of text in the provided chunk, and think about
what would be good, thoughtful, and interesting follow-up questions according to the above rules.

Then you need to provide me with these follow-up questions and also tell me the exact span of text each follow-up
question corresponds to, as in follows up on, in your response. Absolutely make sure that the span you provide is an
exact substring of the chunk text, and that it is not a summary or paraphrase. It should be the exact text corresponding
to the question.

If you can't come up with any follow-up questions at all (which is perfectly fine), just return None.

### REASONING PROCESS
  <Explain your reasoning and thinking process>

### JSON
  {
      "follow_up_questions": [
                          {
                              "question": "<generated follow-up question>",
                              "span": "<exact span of text in the provided chunk corresponding to this question>"
                              "question_type": <Content/Insights/Measurement/Method>,
                              "question_subtype": "<content-clarification/insights-more/insights-nuance/insights-soundne
                              ss/measurement-more/measurement-detail/measurement-alternative/method-alternative/method-d
                              etail/method-motivation>"
                          },

                          ...
                      ] or None
  }

User:

[Start of Context]
{context}
[End of Context]

[[Start of *Provided Chunk*]]
{segment}
[[End of *Provided Chunk*]]
```

B Answerability Prompt

Prompt B

```
System: You are logical, intelligent, precise, and can understand the contents of research papers. You are knowledgable
on different fields and domains of science and engineering. You are able to interpret research papers, and determine
whether information inside these papers can answer some question.

This is your task: Read the following several chunks from a paper and answer whether the question can be answered by one
or more of these chunks. If the question can be answered, report the chunk numbers of the chunks where you found the
answer and answer the question in a sentence or two using only the information in that chunk(s).
```

```
When you respond, first think really hard, step-by-step, about every sentence in every chunk, and analyze whether this
information can answer the question provided.

You can definitely find the question to be unanswerable. You can only find a question to be answerable if ONLY the
information within the provided chunks fully answers this question.

If you find the question to be answerable, please select the fewest possible chunks you would need to answer the
question. If there is one chunk that can fully answer this question, please select only that chunk. If there are multiple
chunks that can fully answer this question, please select only the one chunk you think best answers the question. You
should only select multiple chunks if the only way to fully answer the question is by combining the information in these
chunks.

The chunk numbers are 1-indexed, meaning the first chunk is chunk number 1, the second chunk is chunk number 2, and so
on. When you recieve the chunks, you will know the chunk number as it appears like this: "CHUNK #<chunk_number>". Please
only report the exact number, and only the number, of the chunks you selected. Do not report a chunk number that is not
in the range of possible chunk numbers. The chunk number **hould not be less than 1** nor should it be greater than the
maximum chunk number. This is unforgivable.

In addition to selecting the chunks, you also need to provide the exact span of text within the chunk that answers the
question. This span should be a substring of the chunk text that directly addresses the question. Absolutely make sure
that the span you provide is an exact substring of the chunk text, and that it is not a summary or paraphrase. It should
be the exact text that answers the question.

You need to provide me with an binary Yes or No as the answer to whether the question is answerable by the provided
chunks. If answerable, provide me with the list of chunk numbers corresponding to your selected chunks. I also require
you detail the exact span of text within the chunk that answers the question. If unanswerable, this can be an empty list.

If answerable, provide me with a 1-2 sentence answer. Otherwise, just answer with an empty string.


### REASONING PROCESS
<Explain your reasoning and thinking process>

### JSON
{
  "is_answerable": "<Yes/No>",
  "selected_chunks": [
    {
      "chunk_number": <chunk number>,
      "span": "<exact span of text within the chunk that answers the question>"
    },
    ...
  ],
  "answer": "<1-2 sentence answer, or empty string if unanswerable>"
}

User:

[Start of Provided Chunks]
{answer_batch}
[End of Provided Chunks]

[Start of Question]
{question}
[End of Question]
```

C Hyperparameter Details

We present details on the specific hyperparameters used here. For OpenAI models (`gpt-4o`, `text-embedding-small`, `gpt-5-mini`, `o3`), we use default hyperparameters as specified by the OpenAI API. For `llama-4-maverick-17B-128E-Instruct`, we specify a temperature of 1 and the maximum token produced limited to 4000 tokens.