Backtracing: Retrieving the Cause of the Query

Anonymous ACL submission

Abstract

Many online content portals allow users to ask questions to supplement their understanding (e.g., of lectures). While information retrieval (IR) systems may provide answers for such user queries, they do not directly assist content creators—such as lecturers who want to improve their content—identify segments that *caused* a user to ask those questions. We introduce the task of *backtracing*, in which systems retrieve the text segment that most likely caused a user query. We formalize three real-world domains for which backtracing is important in improving content delivery and communication: understanding the cause of (a) student confusion in the LECTURE domain, (b) reader curiosity in the NEWS ARTICLE domain, and (c) user emotion in the CONVERSATION domain. We evaluate the zero-shot performance of popular information retrieval methods and language modeling methods, including bi-encoder, re-ranking and likelihood-based methods and ChatGPT. While traditional IR systems retrieve semantically relevant information (e.g., details on "projection matrices" for a query "does projecting multiple times still lead to the same point?"), they often miss the causally relevant context (e.g., the lecturer states "projecting twice gets me the same answer as one projection"). Our results show that there is room for improvement on backtracing and it requires new retrieval approaches. We hope our benchmark serves to improve future retrieval systems for backtracing, spawning systems that refine content generation and identify linguistic triggers influencing user queries.

1 Introduction

Content creators and communicators, such as lecturers, greatly value feedback on their content to address confusion and enhance its quality (Evans and Guymon, 1978; Hativa, 1998). For example, when a student is confused by a lecture content, they post questions on the course forum seeking clarification. Lecturers want to determine *where* in the lecture the misunderstanding stems from in order to improve their teaching materials (McKone, 1999; Harvey, 2003; Gormally et al., 2014). The needs of these *content creators* are different than the needs of *information seekers* like students, who may directly rely on information retrieval (IR) systems such as Q&A methods to satisfy their information needs (Schütze et al., 2008; Yang et al., 2015; Rajpurkar et al., 2016; Joshi et al., 2017; Yang et al., 2018).

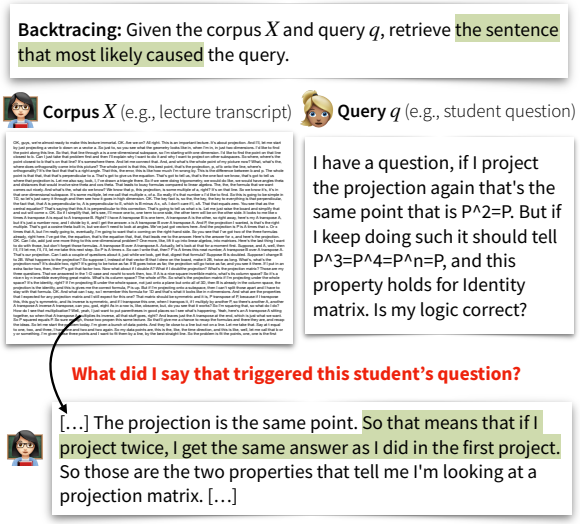

Figure 1: The task of backtracing takes a query and identifies the context that triggers this query. Identifying the cause of a query can be challenging because of the lack of explicit labeling, large corpus size, and domain expertise to understand both the query and corpus.

Identifying the cause of a query can be challenging because of the lack of explicit labeling, implicit nature of additional information need, large size of corpus, and required domain expertise to understand both the query and corpus. Consider the example shown in Figure 1. First, the student does not explicitly flag what part of the lecture causes their question, yet they express a latent need for additional information outside of the lecture con-

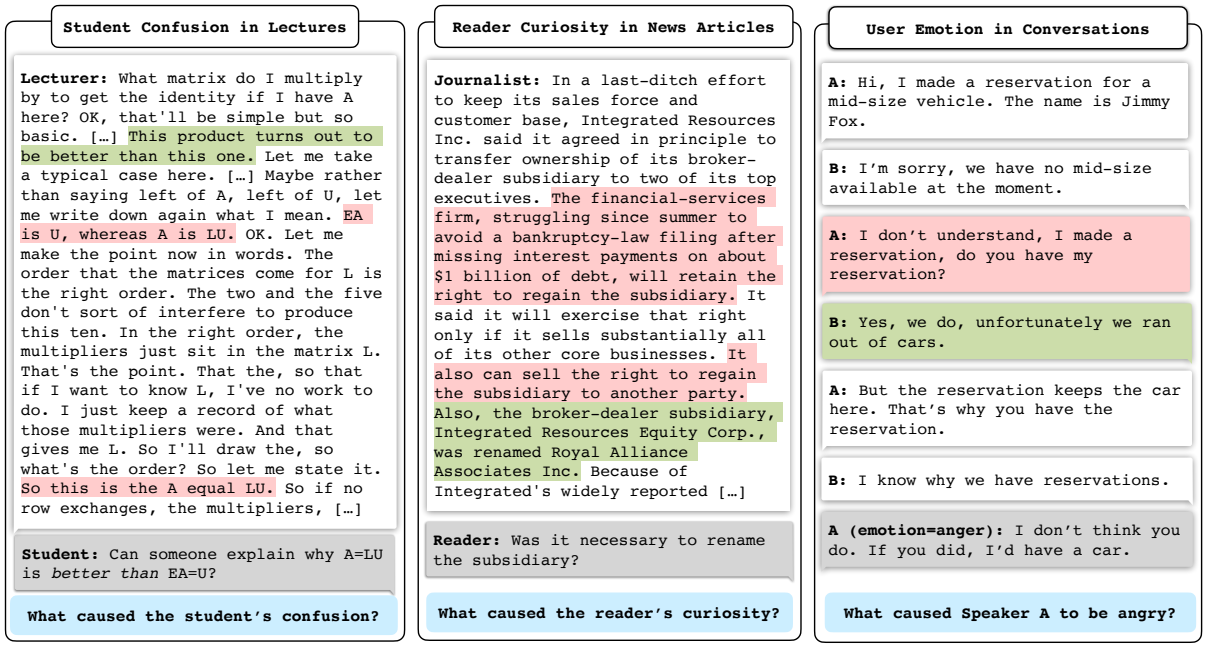

Figure 2: Retrieving the correct triggering context can provide insight into how to better satisfy the user's needs and improve content delivery. We formalize three real-world domains for which backtracing is important in providing context on a user's query: (a) The LECTURE domain where the objective is to retrieve the cause of student confusion; (b) The NEWS ARTICLE domain where the objective is to retrieve the cause of reader curiosity; (c) The CONVERSATION domain where the objective is to retrieve the cause of user emotion (e.g., anger). The user's query is shown in the gray box and the triggering context is the green -highlighted sentence. Popular retrieval systems such as dense retriever-based and re-ranker based systems retrieve incorrect contexts shown in red.

tent. Second, texts like lecture transcripts are long documents; a lecturer would have a difficult time pinpointing the precise source of confusion for every student question they receive. Finally, some queries require domain expertise for understanding the topic and reason behind the student's confusion; not every student question reflects the lecture content verbatim, which is what makes backtracing interesting and challenging.

To formalize this task, we introduce a novel retrieval task called *backtracing*. Given a query (e.g., a student question) and a corpus (e.g., a lecture transcript), the system must identify the sentence that most likely provoked the query. We formalize three real-world domains for which backtracing is important for improving content delivery and communication. First is the LECTURE domain where the goal is to retrieve the cause of student confusion; the query is a student's question and the corpus is the lecturer's transcript. Second is the NEWS ARTICLE domain where the goal is to retrieve the cause of a user's curiosity in the news article domain; the query is a user's question and the corpus is the news article. Third is the CONVERSATION domain where the goal is to retrieve the cause of a user's emotion (e.g., anger); the query is the user's conversation turn expressing that emotion and the corpus is the complete conversation. Figure 2 illustrates an example for each of these domains. These diverse domains showcase the applicability and common challenges of backtracing for improving content generation, similar to heterogeneous IR datasets like BEIR (Thakur et al., 2021).

We evaluate a suite of popular retrieval systems, like dense retriever-based (Reimers and Gurevych, 2019a; Guo et al., 2020; Karpukhin et al., 2020) or re-ranker-based systems (Nogueira and Cho, 2019; Craswell et al., 2020; Ren et al., 2021). Additionally, we evaluate likelihood-based retrieval methods which use pre-trained language models (PLMs) to estimate the probability of the query conditioned on variations of the corpus (Sachan et al., 2022), such as measuring the query likelihood conditioned on the corpus with and without the candidate segment. Finally, we also evaluate the long context window `gpt-3.5-turbo-16k` ChatGPT model because of its ability to process long texts and perform instruction following. We find that there is room for improvement on backtracing across all methods. For example, the bi-encoder systems (Reimers and

Gurevych, 2019a) struggle when the query is not semantically similar to the text segment that causes it; this often happens in the CONVERSATION and LECTURE domain, where the query may be phrased differently than the original content. Overall, our results indicate that backtracing is a challenging task which requires new retrieval approaches to take in *causal* relevance into account; for instance, the top-3 accuracy of the best model is only 44% on the LECTURE domain.

In summary, we make the following contributions in this paper:

- We propose a new task called backtracing where the goal is to retrieve the cause of the query from a corpus. This task targets the information need of *content creators* who wish to improve their content in light of questions from *information seekers*.
- We formalize a benchmark consisting of three domains for which backtracing plays an important role in identifying the context triggering a user's query: retrieving the cause of student confusion in the LECTURE setting, reader curiosity in the NEWS ARTICLE setting, and user emotion in the CONVERSATION setting.
- We evaluate a suite of popular retrieval systems, including bi-encoder and re-ranking architectures, as well as likelihood-based methods that use pretrained language models to estimate the probability of the query conditioned on variations of the corpus.
- We show that there is room for improvement and limitations in current retrieval methods for performing backtracing, suggesting that the task is not only challenging but also requires new retrieval approaches.

2 Related works

The task of information retrieval (IR) aims to retrieve relevant documents or passages that satisfy the information need of a user (Schütze et al., 2008; Thakur et al., 2021). Prior IR techniques involve neural retrieval methods like ranking models (Guo et al., 2016; Xiong et al., 2017; Khattab and Zaharia, 2020) and representation-focused language models (Peters et al., 2018; Devlin et al., 2018; Reimers and Gurevych, 2019a). Recent works also use PLMs for ranking texts in performing retrieval (Zhuang and Zuccon, 2021; Zhuang et al., 2021; Sachan et al., 2022); an advantage of using PLMs is not requiring any domain- or task-specific training, which is useful for settings where there is not enough data for training new models. These approaches have made significant advancements in assisting *information seekers* in accessing information on a range of tasks. Examples of these tasks include recommending news articles to read for a user in the context of the current article they're reading (Voorhees, 2005; Soboroff et al., 2018), retrieving relevant bio-medical articles to satisfy health-related concerns (Tsatsaronis et al., 2015; Boteva et al., 2016; Roberts et al., 2021; Soboroff, 2021), finding relevant academic articles to accelerate a researcher's literature search (Voorhees et al., 2021), or extracting answers from texts to address questions (Yang et al., 2015; Rajpurkar et al., 2016; Joshi et al., 2017; Yang et al., 2018).

However, the converse needs of *content creators* have received less exploration. For instance, understanding what aspects of a lecture cause students to be confused remains under-explored and marks areas for improvement for content creators. Backtracing is related to work on predicting search intents from previous user browsing behavior for understanding why users issue queries in the first place and what trigger their information needs (Cheng et al., 2010; Kong et al., 2015; Koskela et al., 2018). The key difference between our approach and prior works is the nature of the input data and prediction task. While previous methods rely on observable user browsing patterns (e.g., visited URLs and click behaviors) for ranking future search results, our backtracing framework leverages the language in the content itself as the context for the user query and the output space for prediction. This shift in perspective allows content creators to get granular insights into specific contextual, linguistic triggers that influence user queries, as opposed to behavioral patterns.

3 Backtracing

Formally, we define backtracing as: Given corpus of N sentences $X = \{x_1, \ldots, x_N\}$ and query q, backtracing selects

$$\hat{t} = \arg\max_{t \in 1 \ldots N} p(t | x_1, \ldots, x_N, q) \qquad (1)$$

where x_t is the t^{th} sentence in corpus X and p is a probability distribution over the corpus indices,

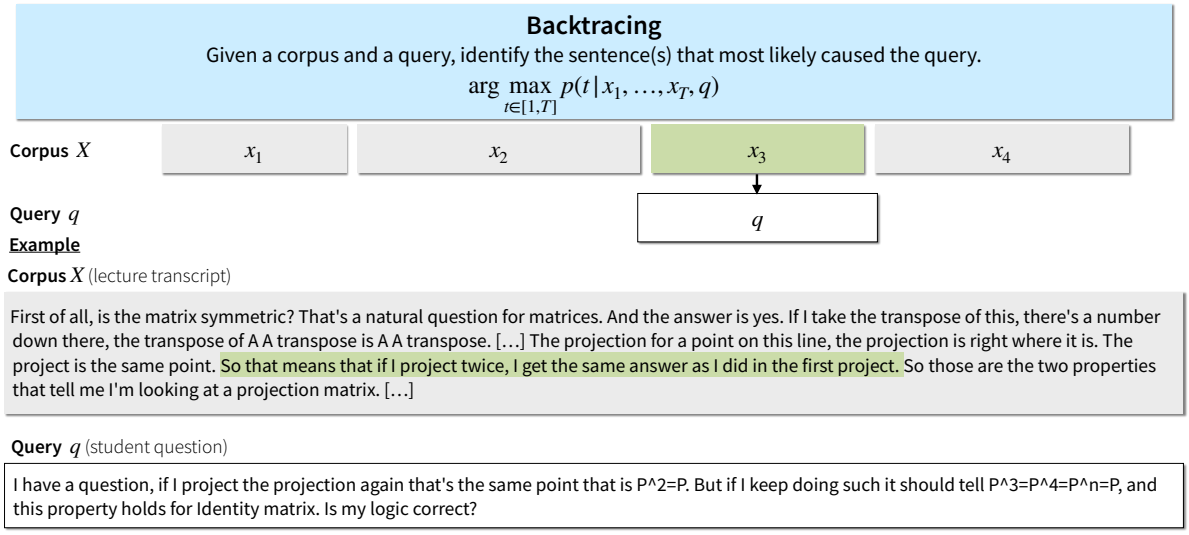

Figure 3: Illustration of backtracing. The goal of backtracing is to identify the most likely sentence from the ordered corpus X that caused the query q. One example is the LECTURE domain where the corpus is a lecture transcript and the query is a student question. The lecturer only discusses about projecting twice and the student further extends that idea to something not raised in the lecture, namely into projecting a matrix an arbitrary n times.

given the corpus and the query. Figure 3 illustrates this definition and grounds it in our previous lecture domain example. This task intuitively translates to: Given a lecture transcript and student question, retrieve the lecture sentence(s) that most likely caused the student to ask that question.

Ideal methods for backtracing are ones that can provide a continuous scoring metric over the corpus and can handle long texts. This allows for distinguishable contributions from multiple sentences in the corpus, as there can be more than one sentence that could cause the query. In the case where there is more than one target sentence, our acceptance criterion is whether there's overlap between the target sentences and the predicted sentence. Additionally, some text domains such as lectures are longer than the context window lengths of existing language models. Effective methods must be able to circumvent this constraint algorithmically (e.g., by repeated invocation of a language model).

Our work explores the backtracing task in a "zero-shot" manner across a variety of domains, similar to Thakur et al. (2021). We focus on a restricted definition of zero-shot in which validation on a small development set is permitted, but not updating model weights. This mirrors many emerging real-world scenarios in which some data-driven interventions can be applied but not enough data is present for training new models. Completely blind zero-shot testing is notoriously hard to conduct within a reusable benchmark (Fuhr, 2018; Perez et al., 2021) and is much less conducive to developing different methods, and thus lies outside our scope.

		LEC	NEWS	CONV
Query	Total	210	1382	671
	Avg. words	30.9	7.1	11.6
	Max words	233	27	62
	Min words	4	1	1
Corpus	Total	11042	2125	8263
	Avg. size	525.8	19.0	12.3
	Max size	948	45	6110
	Min size	273	7	6

Table 1: Dataset statistics on the query and corpus sizes for backtracing. LEC is the LECTURE domain, NEWS is the NEWS ARTICLE domain, and CONV is the CONVERSATION domain. The corpus size is measured on the level of sentences for LECTURE and NEWS ARTICLE, and of conversation turns for CONVERSATION.

4 Backtracing Benchmark Domains

We use a diverse set of domains to establish a benchmark for backtracing, highlighting both its broad applicability and the shared challenges inherent to the task. This section first describes the domain datasets and then describes the dataset statistics with respect to the backtracing task.

4.1 Domains

Figure 2 illustrates examples of the corpus and query in each domain. Table 1 contains statistics on the dataset. The datasets are protected under the CC-BY license.

LECTURE We use real-world university lecture transcripts and student comments to construct the LECTURE domain. Lectures are a natural setting for students to ask questions to express confusion about novel concepts. Lecturers can benefit from knowing what parts of their lecture cause confusion. We adapt the paired comment-lecture dataset from SIGHT (Wang et al., 2023), which contains lecture transcripts from MIT OpenCourseWare math videos and real user comments from YouTube expressing confusion. While these comments naturally act as queries in the backtracing framework, the comments do not have ground-truth target annotations on what *caused* the comment in the first place. Our work contributes these annotations. Two annotators (co-authors of this paper) familiar with the task of backtracing and fluent in the math topics at a university-level annotate the queries[1]. They select up to 5 sentences and are allowed to use the corresponding video to perform the task. 20 queries are annotated by both annotators and these annotations share high agreement: the annotators identified the same target sentences for 70% of the queries, and picked target sentences close to each other. *These annotation results indicate that performing backtracing with consensus is possible.* Appendix B includes more detail on the annotation interface and agreement. The final dataset contains 210 annotated examples, comparable to other IR datasets (Craswell et al., 2020, 2021; Soboroff, 2021).[2] In the case where a query has more than one target sentence, the accuracy criterion is whether there's overlap between the target sentences and predicted sentence (see task definition in Section 3).

NEWS ARTICLE We use real-world news articles and questions written by crowdworkers as they read through the articles to construct the NEWS ARTICLE domain. News articles are a natural setting for readers to ask curiosity questions, expressing a need for more information. We adapt the dataset from Ko et al. (2020) which contains news articles and questions indexed by the article sentences that provoked curiosity in the reader. We modify the dataset by filtering out articles that cannot fit within the smallest context window of models used in the likelihood-based retrieval methods (i.e., 1024 tokens). This adapted dataset allows us to assess the ability of methods to incorporate more contextual information and handling more distractor sentences, while maintaining a manageable length of text. The final dataset contains 1382 examples.

CONVERSATION We use two-person conversations which have been annotated with emotions, such as *anger* and *fear*, and cause of emotion on the level of conversation turns. Conversations are natural settings for human interaction where a speaker may accidentally say something that evokes strong emotions like anger. These emotions may arise from cumulative or non-adjacent interactions, such as the example in Figure 2. Identifying utterances that elicit certain emotions can pave the way for better emotional intelligence in systems and refined conflict resolution tools. We adapt the conversation dataset from Poria et al. (2021) which contain turn-level annotations for the emotion and its cause, and is designed for recognizing the cause of emotions. The query is one of the speaker's conversation turn annotated with an emotion and the corpus is all of the conversation turns. To ensure there are enough distractor sentences, we use conversations with at least 5 sentences and use the last annotated utterance in the conversation. The final dataset contains 671 examples.

4.2 Domain Analysis

To contextualize the experimental findings in Section 6, we first analyze the structural attributes of our datasets in relation to backtracing.

How similar is the query to the cause? To answer this question, we plot the semantic similarity of the query to the ground-truth cause sentence (GT) in Figure 4. We additionally plot the maximal similarity of the query to any corpus sentence (Max) and the difference between the ground-truth and maximal similarity (Diff). This compares the distractor sentences to the ground-truth sentences; the larger the difference is, the less likely semantic relevance can be used as a proxy for *causal* relevance needed to perform backtracing. This would also indicate that poor performance of similarity-based methods because the distractor sentences exhibit higher similarity. We use the `all-MiniLM-L12-v2` S-BERT model to measure semantic similarity (Reimers and Gurevych, 2019a).

[1]The annotators must be fluent in the math topics to understand both the lecture and query, and backtrace accordingly.

[2]After conducting 2-means 2-sided equality power analysis, we additionally concluded that the dataset size is sufficiently large—the analysis indicated a need for 120 samples to establish statistically significant results, with power $1 - \beta = 0.8$ and $\alpha = 0.05$.

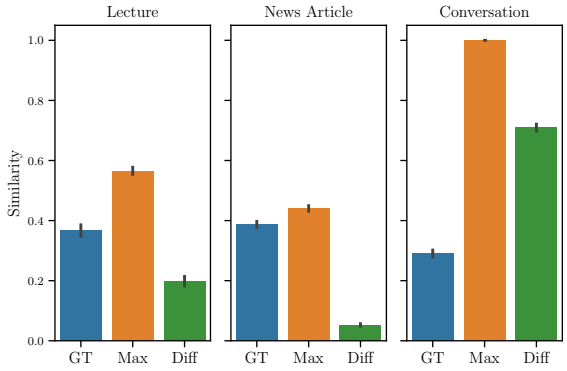

Figure 4: Each dataset plot shows the query similarity to the ground truth cause sentence (GT), to the corpus sentence with maximal similarity (Max), and the difference between the maximal and ground-truth similarity sentences (Diff).

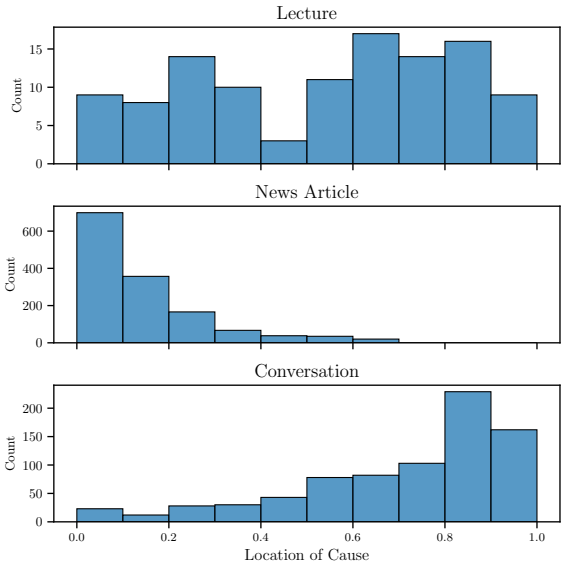

Figure 5: Each row plot is a per-domain histogram of where the ground-truth cause sentence lies in the corpus document. The x-axis reports the location of the cause sentence; 0 means the cause sentence is the first sentence and 1 the last sentence. The y-axis reports the count of cause sentences at that location.

Lecture: [...] So it's 1 by 2x0 times 2y0, which is 2x0y0, which is, lo and behold, 2. **[...]**
Student A's question: why is 2xo(yo) = 2?
Student B's question: When he solves for the area of the triangle, why does he say it doesn't matter what X0 and Y0 are? Does he just mean that all values of f(x) = 1/x will result in the area of the triangle of the tangent line to be 2?
Student C's question: Why always 2?? is there a prove?

Figure 6: An example of a common confusion point where several students posed questions concerning a particular part of the lecture.

Notably, the queries and their ground-truth cause sentences exhibit low semantic similarity across domains, indicated by the low blue bars. Additionally, indicated by the green bars, CONVERSATION and LECTURE have the largest differences between the ground-truth and maximal similarity sentences, whereas NEWS ARTICLE has the smallest. This suggests that there may be multiple passages in a given document that share a surface-level resemblance with the query, but a majority do not cause the query in the CONVERSATION and LECTURE domains. In the NEWS ARTICLE domain, the query and cause sentence exhibit higher semantic similarity because the queries are typically short and mention the event or noun of interest. Altogether, this analysis brings forth a key insight: Semantic relevance doesn't always equate causal relevance.

Where are the causes located in the corpus? Understanding the location of the cause provides insight into how much context is needed in identifying the cause to the query. Figure 5 visualizes the distribution of cause sentence locations within the corpus documents. These plots show that while some domains have causes concentrated in specific sections, others exhibit a more spread-out pattern. For the NEWS ARTICLE domain, there is a noticeable peak at the beginning of the documents which suggests little context is needed to identify the cause. This aligns with the typical structure of news articles where crucial information is introduced early to capture the reader's interest. As a result, readers may have immediate questions from the onset. Conversely, in the CONVERSATION domain, the distribution peaks at the end, suggesting that more context from the conversation is needed to identify the cause. Finally, in the LECTURE domain, the distribution is relatively uniform which suggests a broader contextual dependence. The causes of confusion arise from any section, emphasizing the importance of consistent clarity throughout an educational delivery.

An interesting qualitative observation is that there are shared cause locations for different queries. An example from the LECTURE domain is shown in Figure 6 where different student questions are mapped to the same cause sentence. This shows the potential for models to effectively perform backtracing and automatically identify common locations of confusion for lecturers to revise for future course offerings.

5 Methods

We evaluate a suite of existing, state-of-the-art retrieval methods and report their top-1 and top-3 accuracies (i.e., whether the top 1 and 3 candidate sentences include the ground-truth sentences). They can be broadly categorized into similarity-based

(i.e., using sentence similarity) and likelihood-based retrieval methods. Similar to Sachan et al. (2022), the likelihood-based retrieval methods use PLMs to measure the probability of the query conditioned on variations of the corpus and can be more expressive than the similarity-based retrieval methods; we describe these variations in detail below. We use GPT-2 (Radford et al., 2019), GPT-J (Wang and Komatsuzaki, 2021), and OPT-6.7B (Zhang et al., 2022) as the PLMs. We additionally evaluate with `gpt-3.5-turbo-16k`, a new model that has a long context window ideal for long text settings like SIGHT. However, because this model does not output probability scores, we cast only report its top 1 accuracy.

Random. This method randomly retrieves a sentence from the corpus.

Edit distance. This method retrieves the sentence with the smallest edit distance from the query.

Bi-encoders. This method retrieves the sentence with the highest semantic similarity using the best performing S-BERT models (Reimers and Gurevych, 2019b). We use `multi-qa-MiniLM-L6-cos-v1` trained on a large set of question-answer pairs and `all-MiniLM-L12-v2` trained on a diversity of text pairs from sentence-transformers as the encoders.

Cross-encoder. This method picks the sentence with the highest predicted similarity score by the cross-encoder. We use `ms-marco-MiniLM-L-6-v2` (Thakur et al., 2021).

Re-ranker. This method uses a bi-encoder to retrieve the top k candidate sentences from the corpus, then uses a cross-encoder to re-rank the k sentences. We use `all-MiniLM-L12-v2` as the bi-encoder and `ms-marco-MiniLM-L-6-v2` as the cross-encoder. Since the smallest dataset—Daily Dialog—has a minimum of 5 sentences, we use $k = 5$ for all datasets.

`gpt-3.5-turbo-16k`. This method is provided a line-numbered corpus and the query, and generates the line number that most likely caused the query. The prompt used for `gpt-3.5-turbo-16k` is in Appendix C.

Single-sentence likelihood-based retrieval $p(q|x_t)$. This method retrieves the sentence $x_t \in X$ that maximizes $p(q|x_t)$. To contextualize the corpus and query, we add domain-specific prefixes to the corpus and query. For example, in SIGHT, we prepend "Teacher says: " to the corpus sentence and "Student asks: " to the query. Due to space constraints, Appendix C contains all the prefixes used.

Auto-regressive likelihood-based retrieval $p(q|x_{\leq t})$. This method retrieves the sentence x_t which maximizes $p(q|x_{\leq t})$. This method evaluates the importance of preceding context in performing backtracing. LECTURE is the only domain where the entire corpus cannot fit into the context window. This means that we cannot always evaluate $p(q|x_{\leq t})$ for x_t when $|x_{\leq t}|$ is longer than the context window limit. For this reason, we split the corpus X into chunks of k sentences, (i.e., $X_{0:k-1}, X_{k:2k-1}, \dots$) and evaluate each x_t within their respective chunk. For example, if $x_t \in X_{k:2k-1}$, the auto-regressive likelihood score for x_t is $p(q|X_{k:t})$. We evaluate with $k = 20$ because it is the maximum number of sentences (in addition to the query) that can fit in the smallest model context window.

Average Treatment Effect (ATE) likelihood-based retrieval $p(q|X) - p(q|X \setminus x_t)$. This method takes inspiration from treatment effects in causal inference (Holland, 1986). We describe how ATE can be used as a retrieval criterion. In our setting, the treatment is whether the sentence x_t is included in the corpus. We're interested in the effect the treatment has on the query likelihood:

$$\texttt{ATE}(x_t) = p_\theta(q|X) - p_\theta(q|X \setminus \{x_t\}). \quad (2)$$

ATE likelihood methods retrieve the sentence that maximizes $\texttt{ATE}(x_t)$. These are the sentences that have the largest effect on the query's likelihood. We directly select the sentences that maximize Equation 2 for NEWS ARTICLE and CONVERSATION. We perform the same text chunking for LECTURE as in the auto-regressive retrieval method: If $x_t \in X_{k:2k-1}$, the ATE likelihood score for x_t is measured as $p(q|X_{k:2k-1}) - p(q|X_{k:2k-1} \setminus \{x_t\})$.

6 Results

The model results are summarized in Table 2.

The best-performing models achieve modest accuracies. For example, on the LECTURE domain with many distractor sentences, the best-performing model only achieves top-3 44% accu-

		LECTURE		NEWS ARTICLE		CONVERSATION	
		@1	@3	@1	@3	@1	@3
	Random	0	0	7	21	12	36
	Edit	4	8	7	18	1	16
	Bi-Encoder (Q&A)	23	37	48	71	1	15
	Bi-Encoder (`all-MiniLM`)	26	40	49	75	1	37
	Cross-Encoder	22	39	66	**85**	1	15
	Re-ranker	29	44	66	**85**	1	21
	`gpt-3.5-turbo-16k`	15	N/A	**67**	N/A	**47**	N/A
Single-sentence $p(q\|s_t)$	GPT2	20	34	43	64	3	46
	GPTJ	23	42	**67**	**85**	5	**65**
	OPT 6B	**30**	**43**	66	82	2	56
Autoregressive $p(q\|s_{\leq t})$	GPT2	11	16	9	18	5	54
	GPTJ	14	24	55	76	8	60
	OPT 6B	16	26	52	73	18	**65**
ATE $p(q\|S) - p(q\|S/\{s_t\})$	GPT2	13	21	51	68	2	24
	GPTJ	8	18	**67**	79	3	18
	OPT 6B	9	20	64	76	3	22

Table 2: Accuracy in percentage (%). The best models in each column are bolded. For each dataset, we report the top-1 and 3 accuracies. `gpt-3.5-turbo-16k` reports N/A for top-3 accuracy because it does not output deterministic continuous scores for ranking sentences.

racy. On the CONVERSATION domain with few distractor sentences, the best-performing model only achieves top-3 65% accuracy. This underscores that measuring causal relevance is challenging and markedly different from existing retrieval tasks.

No model performs consistently across domains. For instance, while a similarity-based method like the Bi-Encoder (`all-MiniLM`) performs well on the NEWS ARTICLE domain with top-3 75% accuracy, it only manages top-3 37% accuracy on the CONVERSATION domain. These results complement the takeaway from the domain analysis in Section 4 that semantic relevance is not a reliable proxy for causal relevance. Interestingly, on the long document domain LECTURE, the long-context model `gpt-3.5-turbo-16k` performs worse than non-contextual methods like single-sentence likelihood methods. This suggests that accounting for context is challenging for current models.

Single-sentence methods generally outperform their autoregressive counterparts except on CONVERSATION. This result complements the observations made in Section 4's domain analysis where the location of the causes concentrates at the start for NEWS ARTICLE and uniformly for LECTURE, suggesting that little context is needed to identify the cause. Conversely, conversations require more context to distinguish the triggering contexts, which suggests why the autoregressive methods perform generally better than the single-sentence methods.

ATE likelihood methods does not signicantly improve upon other methods. Even though the ATE likelihood method is designed the calculate the effect of the cause sentence, it competes with noncontextual methods such as the single-sentence likelihood methods. This suggest challenges in using likelihood methods to measure the counterfactual effect of a sentence on a query.

7 Conclusion

In this paper, we introduce the novel task of backtracing, which aims to retrieve the text segment that most likely provokes a query. This task addresses the information need of *content creators* who want to improve their content, in light of queries from information seekers. We introduce a benchmark that covers a variety of domains, such as the news article and lecture setting. We evaluate a series of methods including popular IR methods, likelihood-based retrieval methods and `gpt-3.5-turbo-16k`. Our results indicate that there is room for improvement across existing retrieval methods. These results suggest that backtracing is a challenging task that requires new retrieval approaches with better contextual understanding and reasoning about causal relevance. We hope our benchmark serves as a foundation for improving future retrieval systems for backtracing, and ultimately, spawns systems that empower content creators to understand user queries, refine their content and provide users with better experiences.

Limitations

Single-sentence focus. Our approach primarily focuses on identifying the most likely single sentence that caused a given query. However, in certain scenarios, the query might depend on groups or combinations of sentences. Ignoring such dependencies can limit the accuracy of the methods.

Content creators in other domains. Our evaluation primarily focuses on the dialog, new article and lecture settings. While these domains offer valuable insights, the performance of backtracing methods may vary in other contexts, such as scientific articles and queries from reviewers. Future work should explore the generalizability of backtracing methods across a broader range of domains and data sources.

Long text settings. Due to the length of the lecture transcripts, the transcripts had to be divided and passed into the likelihood-based retrieval methods. This approach may result in the omission of crucial context present in the full transcript, potentially affecting the accuracy of the likelihood-based retrieval methods. Exploring techniques to effectively handle larger texts and overcome model capacity constraints would be beneficial for improving backtracing performance in long text settings, where we would imagine backtracing to be useful in providing feedback for.

Multimodal sources. Our approach identifies the most likely text segment in a corpus that caused a given query. However, in multimodal settings, a query may also be caused by other data types, e.g., visual cues taht are not captured in the transcripts. Ignoring such non-textual data can limit the accuracy of the methods.

Ethics Statement

Empowering content creators to refine their content based on user feedback contributes to the production of more informative materials. Therefore, our research has the potential to enhance the educational experiences of a user, by assisting content creators through backtracing. Nonetheless, we are mindful of potential biases or unintended consequences that may arise through our work and future work. For example, the current benchmark analyzes the accuracy of backtracing on English datasets and uses PLMs trained predominantly on English texts. As a result, the inferences drawn from the current backtracing results or benchmark may not accurately capture the causes of multilingual queries, and should be interpreted with caution.

References

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, pages 716–722. Springer.

Zhicong Cheng, Bin Gao, and Tie-Yan Liu. 2010. Actively predicting diverse search intent from user browsing behaviors. In *Proceedings of the 19th international conference on World wide web*, pages 221–230.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the trec 2020 deep learning track.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the trec 2019 deep learning track.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Warren E Evans and Ronald E Guymon. 1978. Clarity of explanation: A powerful indicator of teacher effectiveness.

Norbert Fuhr. 2018. Some common mistakes in ir evaluation, and how they can be avoided. In *Acm sigir forum*, volume 51, pages 32–41. ACM New York, NY, USA.

Cara Gormally, Mara Evans, and Peggy Brickman. 2014. Feedback about teaching in higher ed: Neglected opportunities to promote change. *CBE—Life Sciences Education*, 13(2):187–199.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 55–64.

Mandy Guo, Yinfei Yang, Daniel Cer, Qinlan Shen, and Noah Constant. 2020. Multireqa: A cross-domain evaluation for retrieval question answering models.

Lee Harvey. 2003. Student feedback [1]. *Quality in higher education*, 9(1):3–20.

Nira Hativa. 1998. Lack of clarity in university teaching: A case study. *Higher Education*, pages 353–381.

Paul W Holland. 1986. Statistics and causal inference. *Journal of the American statistical Association*, 81(396):945–960.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

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.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Wei-Jen Ko, Te-Yuan Chen, Yiyan Huang, Greg Durrett, and Junyi Jessy Li. 2020. Inquisitive question generation for high level text comprehension. *arXiv preprint arXiv:2010.01657*.

Weize Kong, Rui Li, Jie Luo, Aston Zhang, Yi Chang, and James Allan. 2015. Predicting search intent based on pre-search context. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 503–512.

Markus Koskela, Petri Luukkonen, Tuukka Ruotsalo, Mats Sjöberg, and Patrik Floréen. 2018. Proactive information retrieval by capturing search intent from primary task context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(3):1–25.

MiniChain Library. 2023. MiniChain Library. `https://github.com/srush/minichain#typed-prompts`. [Online; accessed 4-June-2024].

Ian McKenzie. 2023. Inverse Scaling Prize: First Round Winners. `https://irmckenzie.co.uk/round1#:~:text=model%20should%20answer.-,Using%20newlines,-We%20saw%20many`. [Online; accessed 4-June-2024].

Kathleen E McKone. 1999. Analysis of student feedback improves instructor effectiveness. *Journal of Management Education*, 23(4):396–415.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in neural information processing systems*, 34:11054–11070.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu Bai Jian, Pengfei Hong, Romila Ghosh, Abhinaba Roy, Niyati Chhaya, et al. 2021. Recognizing emotion cause in conversations. *Cognitive Computation*, 13:1317–1332.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Nils Reimers and Iryna Gurevych. 2019a. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Nils Reimers and Iryna Gurevych. 2019b. Sentence-bert: Sentence embeddings using siamese bert-networks.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. *arXiv preprint arXiv:2110.07367*.

Kirk Roberts, Dina Demner-Fushman, Ellen M Voorhees, Steven Bedrick, and Willian R Hersh. 2021. Overview of the trec 2021 clinical trials track. In *Proceedings of the Thirtieth Text REtrieval Conference (TREC 2021)*.

Devendra Singh Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. *arXiv preprint arXiv:2204.07496*.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

Ian Soboroff. 2021. Overview of trec 2021. In *30th Text REtrieval Conference. Gaithersburg, Maryland*.

Ian Soboroff, Shudong Huang, and Donna Harman. 2018. Trec 2018 news track overview. In *TREC*, volume 409, page 410.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):1–28.

Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2021. Trec-covid: constructing a pandemic information retrieval test collection. In *ACM SIGIR Forum*, volume 54, pages 1–12. ACM New York, NY, USA.

Ellen M Voorhees. 2005. The trec robust retrieval track. In *ACM SIGIR Forum*, volume 39, pages 11–20. ACM New York, NY, USA.

Ben Wang and Aran Komatsuzaki. 2021. Gpt-j-6b: A 6 billion parameter autoregressive language model.

Rose Wang, Pawan Wirawarn, Noah Goodman, and Dorottya Demszky. 2023. Sight: A large annotated dataset on student insights gathered from higher education transcripts. In *Proceedings of Innovative Use of NLP for Building Educational Applications*.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pages 55–64.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Shengyao Zhuang, Hang Li, and Guido Zuccon. 2021. Deep query likelihood model for information retrieval. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*, pages 463–470. Springer.

Shengyao Zhuang and Guido Zuccon. 2021. Tilde: Term independent likelihood model for passage reranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1483–1492.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2023. Can large language models transform computational social science? *arXiv preprint arXiv:2305.03514*.

A Computational Setup

We ran our experiments on a Slurm-based university compute cluster, consisting of interconnected nodes optimized for intensive computation tasks and shared among multiple users for research purposes. The experiments varied in length in time—some took less than an hour to run (e.g., the random baselines), while others took a few days to run (e.g., the ATE likelihood-based methods on LECTURE).

B LECTURE annotation interface

Figure 7 shows the interface used for annotating the LECTURE dataset.

C Contextualized prefixes for scoring

This section describes the prompts used for the likelihood-based retrieval methods and `gpt-3.5-turbo-16k`.

The prompts used for `gpt-3.5-turbo-16k` follow the practices in works from NLP, education and social sciences (McKenzie, 2023; Library, 2023; Ziems et al., 2023; Wang et al., 2023). Specifically, we enumerate the sentences in the corpus as multiple-choice options and each option is separated by a newline. We add context for the task at the start of the prompt, and the constraints of outputting a JSON-formatted text for the task at the end of the prompt. We found the model to be reliable in outputting the text in the desirable format.

C.1 LECTURE

For the likelihood-based retrieval methods, the sentences are concatenated by spaces and "A teacher is teaching a class, and a student asks a question.\nTeacher: " is prepended to the corpus. Because the text comes from transcribed audio which is not used in training dataset of the PLMs

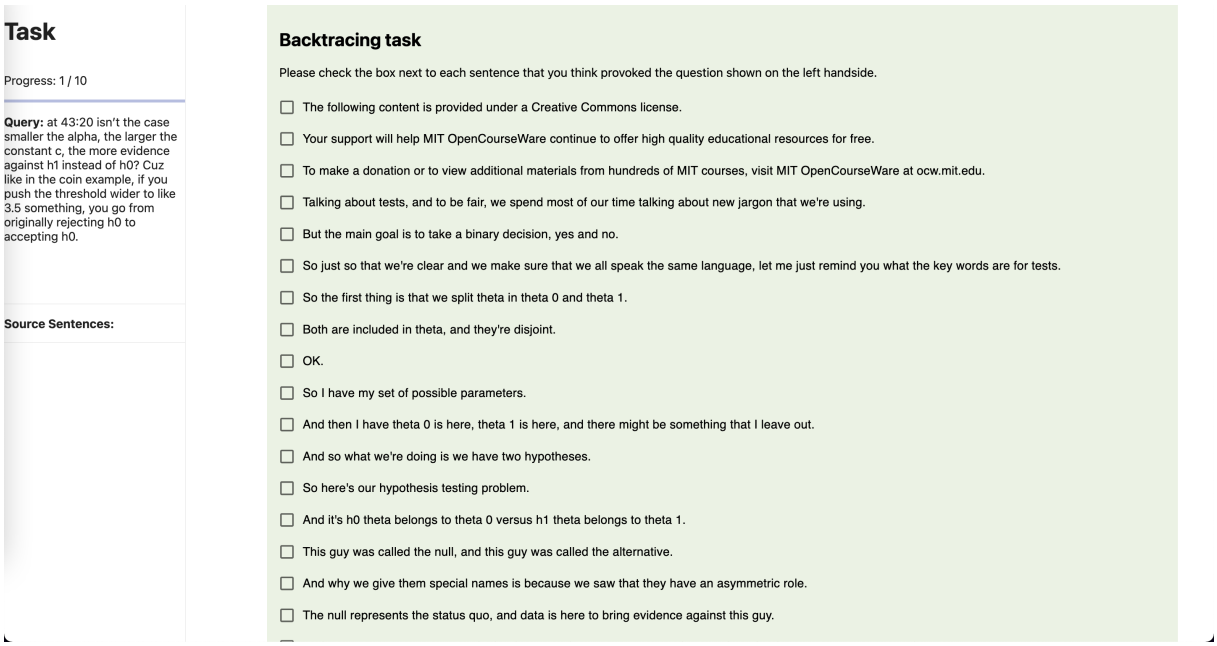

Figure 7: Annotation interface

we use in our work, we found it important for additional context to be added in order for the probabilities to be slightly better calibrated. For the query, "Student: " is prepended to the text. For example, $X =$ "A teacher is teaching a class, and a student asks a question.\n Teacher: [sentence 1] [sentence 2] ...", and $q =$ "Student: [query]".

The prompt used for `gpt-3.5-turbo-16k` is in Figure 8.

C.2 News Article

For the likelihood-based retrieval methods, the sentences are concatenated by spaces and "Text: " is prepended to the corpus. For the query, "Question: " is prepended to the text. For example, $X =$ "Text: [sentence 1] [sentence 2] ...", and $q =$ "Question: [question]".

The prompt used for `gpt-3.5-turbo-16k` is in Figure 9.

C.3 Conversation

For the likelihood-based retrieval methods, the speaker identity is added to the text, and the turns are separated by line breaks. For the query, the same format is used. For example, $X =$ "Speaker A: [utterance]\nSpeaker B: [utterance]", and $q =$ "Speaker A: [query]".

The prompt used for `gpt-3.5-turbo-16k` is in Figure 10.

gpt-3.5-turbo-16k prompt for LECTURE

Consider the following lecture transcript:
{line-numbered transcript}

Now consider the following question:
{query}

Which of the transcript lines most likely provoked this question? If there are multiple possible answers, list them out. Format your answer as: ["line number": integer, "reason": "reason for why this line most likely caused this query", ...]

Figure 8: gpt-3.5-turbo-16k prompt for LECTURE. For the line-numbered transcript, "Teacher: " is prepended to each sentence, the sentences are separated by line breaks, and each line begins with its line number. For the query, "Student: " is prepended to the text. For example, a line-numbered article looks like "0. Teacher: [sentence 1]\n1. Teacher: [sentence 2]\n2. Teacher: [sentence 3] ...", and the query looks like "Student: [query]".

gpt-3.5-turbo-16k prompt for NEWS ARTICLE

Consider the following article:
{line-numbered article}

Now consider the following question:
{query}

Which of the article lines most likely provoked this question? If there are multiple possible answers, list them out. Format your answer as: ["line number": integer, "reason": "reason for why this line most likely caused this query", ...]

Figure 9: gpt-3.5-turbo-16k prompt for NEWS ARTICLE. For the line-numbered article, "Text: " is prepended to each sentence, the sentences are separated by line breaks, and each line begins with its line number. For the query, "Question: " is prepended to the text. For example, a line-numbered article looks like "0. Text: [sentence 1]\n1. Text: [sentence 2]\n2. Text: [sentence 3] ...", and the query looks like "Question: [question]".

gpt-3.5-turbo-16k prompt for CONVERSATION

Consider the following conversation:
{line-numbered conversation}

Now consider the following line:
{query}

The speaker felt {emotion} in this line. Which of the conversation turns (lines) most likely caused this emotion? If there are multiple possible answers, list them out. Format your answer as: ["line number": integer, "reason": "reason for why this line most likely caused this emotion", ...]

Figure 10: gpt-3.5-turbo-16k prompt for CONVERSATION. For the line-numbered conversation, the speaker is added to each turn, the turns are separated by line breaks, and each line begins with its line number. For the query, the speaker is also added. For example, a line-numbered conversation may look like "0. Speaker A: [utterance]\n1. Speaker B: [utterance]\n2. Speaker A: [utterance] ...", and the query may look like "Speaker A: [query]".