KTRL+F: Knowledge-Augmented In-Document Search

Hanseok Oh[1*] **Haebin Shin**[1,2*] **Miyoung Ko**[1] **Hyunji Lee**[1] **Minjoon Seo**[1]
[1]KAIST AI [2]Samsung Research
{hanseok, haebin.shin, miyoungko, hyunji.amy.lee, minjoon}@kaist.ac.kr

Abstract

We introduce a new problem KTRL+F, a knowledge-augmented in-document search that necessitates real-time identification of all semantic targets within a document with the awareness of external sources through a single natural query. KTRL+F addresses following unique challenges for in-document search: 1) utilizing knowledge outside the document for extended use of additional information about targets, and 2) balancing between real-time applicability with the performance. We analyze various baselines in KTRL+F and find limitations of existing models, such as hallucinations, high latency, or difficulties in leveraging external knowledge. Therefore, we propose a Knowledge-Augmented Phrase Retrieval model that shows a promising balance between speed and performance by simply augmenting external knowledge in phrase embedding. We also conduct a user study to verify whether solving KTRL+F can enhance search experience for users. It demonstrates that even with our simple model, users can reduce the time for searching with less queries and reduced extra visits to other sources for collecting evidence. We encourage the research community to work on KTRL+F to enhance more efficient in-document information access.[1]

1 Introduction

Despite significant advancement in many Natural Language Processing applications, facilitated by transformer-based models (Devlin et al., 2019; Raffel et al., 2019), real-time in-document search still leans heavily on conventional lexical matching tools like the "Find" function (Ctrl+F) and regular expressions. These tools, while fast, have clear limitations, especially with ambiguous keywords or multiple targets.

Machine Reading Comprehension (MRC) seems a promising solution to these issues. It reads documents, comprehends their context, and answers questions (Rajpurkar et al., 2016). However, MRC focuses on explicit contents, limiting its value when users need knowledge not directly in the document (Trischler et al., 2017; Rajpurkar et al., 2018; Joshi et al., 2017). Consider a scenario where users read a news article and seek for information on the *"Social network platform of China."* (Figure 1). Typically, users refer to external sources such as Wikipedia to gather additional details not explicitly mentioned in news related to candidates, such as *WeChat, Baidu, and Twitter*. An alternative is harnessing the capabilities of powerful pre-trained language models (Brown et al., 2020; Touvron et al., 2023). However, their generative nature poses challenges for real-time search task.

To overcome the limitations of previous methods and enhance the efficiency and comprehensiveness of in-document search, we present a new problem KTRL+F (knowledge-augmented in-document search). This task aims to reduce redundancy and better meet the requirements of real users. Given a natural language query and a long input document, KTRL+F is designed to fulfill three key criteria: (REQ 1) Find all semantic targets. (REQ 2) Utilizes external knowledge. (REQ 3) Operates in real-time. In the absence of a suitable dataset to evaluate KTRL+F, we curate a new dataset with unique queries demanding matching external evidence. To measure model performance in KTRL+F, we introduce a set of reformulated metrics tailored to measure processing speed while maintaining robust and high performance.

We conduct an extensive analysis of various baselines for KTRL+F and find several limitations including hallucination, slow speed with generative models, and challenges in incorporating external knowledge into MRC models (see §6.2 for details). To strike a balance between real-time processing

[*] indicates equal contribution
[1]Code, Chrome extension plugin, and dataset are available at https://github.com/kaistAI/KtrlF

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 2416–2436

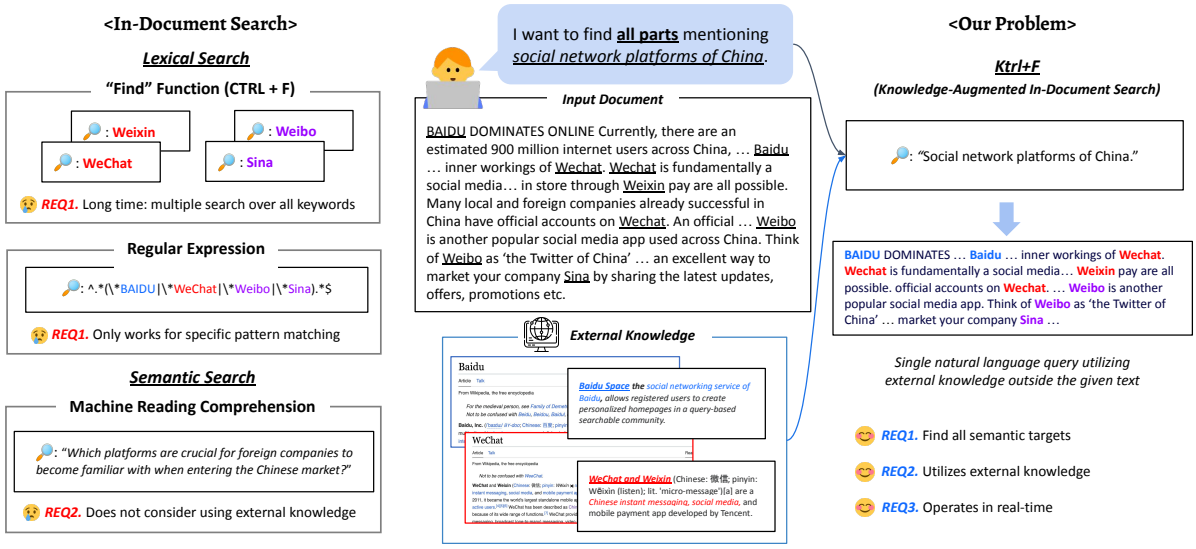


Figure 1: Comparison between in-document search and KTRL+F problem. In-document search accesses the information in documents by either lexical search (Ctrl+F, Regular expression) or semantic search (MRC). Lexical search suffers from finding semantically matching keywords, and semantic search does not consider external knowledge. KTRL+F requires an efficient way to utilize external knowledge to find all semantic targets in real-time.

speed and achieving high performance through effective utilization of additional knowledge, we introduce a simple yet effective extension of phrase retrieval (Lee et al., 2021): Knowledge-Augmented Phrase Retrieval. This model seamlessly extends the phrase retrieval to cater to in-document search scenarios, all while integrating external knowledge without the need for additional training steps. Our experiments support that by simply adding the knowledge embedding and the phrase embedding, Knowledge-Augmented Phrase Retrieval exhibits the potential to reflect external knowledge without sacrificing latency.

Furthermore, we conduct a user study to show the necessity of KTRL+F utilizing a Chrome extension plugin that operates in the real web environments, built upon our model. Results of the study demonstrate that search experience of users can be enhanced even with our simple model with seamless access to external knowledge during in-document searches. We encourage the research community to take on the unique challenge of solving KTRL+F requiring balance between performance and speed to enhance more efficient and effective information access.

2 Related Works

Machine Reading Comprehension (MRC) is a task to find the answer to a question in the provided context. Most MRC datasets assess the ability of context understanding of the model by extracting a single span for the query only grounding on the information within a provided context (Rajpurkar et al., 2016; Trischler et al., 2017; Joshi et al., 2017; Rajpurkar et al., 2018; Fisch et al., 2019; Kwiatkowski et al., 2019). Few works explore the identification of multiple targets for a query in the input document evaluating the model's comprehension of the given context (Dasigi et al., 2019; Zhu et al., 2020; Li et al., 2022) . Some studies tackle information-seeking problem by utilizing external information missing from input document to gap knowledge (Ferguson et al., 2020; Dasigi et al., 2021). This external information aids in enhancing the understanding of the context. However, since the KTRL+F relies on external knowledge beyond its context, it is essential to explicitly ground external knowledge about the target. Consequently, the evaluation of KTRL+F focuses not on the understanding of the given context, but on information obtained from outside the given context.

Knowledge-augmented information retrieval is an approach to enrich external information within the text embedding. The introduction of a knowledge-augmented design aims to supplement deficient contextual information, thereby enhancing the richness of text embedding. Numerous studies tackle knowledge augmentation across various NLP tasks (Zhang et al., 2019; Xiong et al., 2019; Peters et al., 2019; Poerner et al., 2020; Févry

et al., 2020; Levine et al., 2020; Wang et al., 2021; Bertsch et al., 2023). The integration of information from diverse sources leads to an improved language understanding ability. However, the application of knowledge augmentation in information retrieval tasks has received comparatively less attention. Lin et al. (2022) attempts to improve text embeddings for retrieval by enriching context information through embeddings derived from a given context, without specifically focusing on external knowledge. Meanwhile, Lee et al. (2023) utilizes contextualized embeddings as vocabulary embeddings for text tokens in a generative retriever, thereby enhancing contextual information for basic text tokens. Additionally, Raina et al. (2023) focuses on the retrieval augmented text embedding to efficiently reuse prebuilt dense representation with lightweight representation, and also discusses the necessity of systems for utilizing external contextual information to include contextual information outside the given context in text embedding tasks. In contrast to these approaches, KTRL+F directs its attention on augmenting knowledge from external sources for entities in a novel in-document retrieval task. This involves extracting information not present in the given text, thus expanding the capabilities of the information retrieval process.

3 Ktrl+F: Knowledge-Augmented In-Document Search

In this section, we define KTRL+F, which is knowledge-augmented in-document search task and its unique characteristics (§3.1). Then we describe the evaluation metrics to measure each requirement (§3.2).

3.1 Task Definition

KTRL+F is a task that requires finding all semantic targets from a given input document in real-time with the awareness of external knowledge, when given a natural language query. As illustrated in Figure 1, when presented with a natural language query and a input document, Ktrl+F is designed to meet three essential criteria.

REQ 1: Find all semantic targets. KTRL+F requires finding all relevant targets within a given document. The term "all" refers to multiple aspects: finding all multiple answers (baidu, wechat, weibo), all occurrences of each answer (baidu appears two times in the document), and all lexical variations of mentions for each answer (Weibo, Sina).

REQ 2: Utilize external knowledge. Expanding the matching space from lexical to semantic introduces a comprehensive connection between query and target units. However, in many cases, targets contain extra information beyond the input document. By effectively leveraging this additional information through utilization of external knowledge, we can further bridge the semantic gap between the query and the targets.

REQ 3: Search in real-time. KTRL+F inherits the practicality of in-document search, such as Ctrl+F, which emphasizes real-time search to minimize the time on finding targets within the input document. The complexity of KTRL+F lies in effectively balancing real-time applicability with the performance of finding all matching targets by leveraging external knowledge.

3.2 Evaluation Metrics

To assess various aspects of KTRL+F, we employ a range of metrics that collectively measure the overall balance of performance and speed. Following Izacard and Grave (2021), we indirectly assess the impact of utilizing external knowledge by comparing the overall performance of the system with and without its incorporation, given the absence of a definite gold standard answer (REQ 2).

List EM F1, List Overlap F1, Robustness Score. The three metrics measure if the model finds all semantic targets, which fulfills REQ 1. List EM considers correct only when the prediction list is exactly the same as the ground truth list. Note that List EM is different from Set EM, a commonly used metric in Machine Reading Comprehension (Rajpurkar et al., 2016), in that List EM aims to identify all occurrences of targets within a input document. Whereas, List Overlap allows partial matches between individual elements of the predicted and the ground truth list, extending set-based partial match from MultispanQA (Li et al., 2022). For detailed equations and explanation for List Overlap, please refer to Appendix I.

Inspired by Zhong et al. (2023), we adjust robustness score to assess the robustness of system in predicting target answer entities as queries change within a given input document. Treating queries linked to the same document as a cohesive cluster, we calculate the robustness score by averaging the minimum score within each cluster. This approach enhances the comprehensive evaluation of KTRL+F task, given that the knowledge-

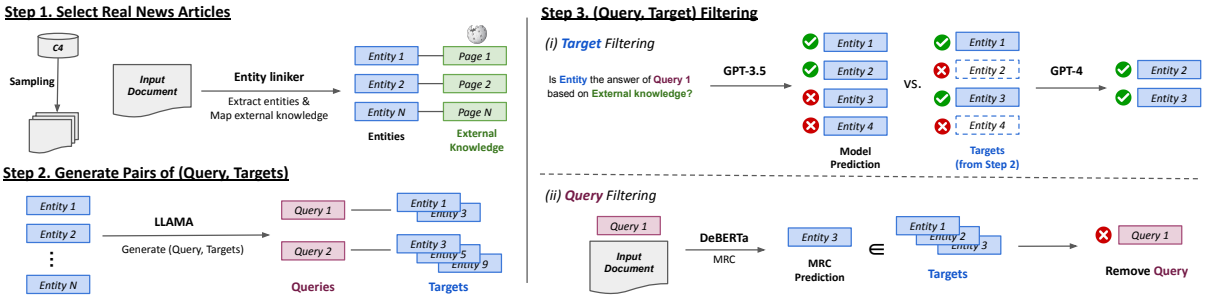


Figure 2: Overview of KTRL+F dataset construction pipeline. We utilize real news articles as input documents (Step 1), and automatically generate queries and targets using LLAMA (Step 2). To enhance the reliability of the identified targets, each entity is re-verified with external knowledge and finalized in (Step 3-1). Additionally, we use the MRC model to eliminate queries that do not meet the criteria outlined in REQ 2 (Step 3-2).

augmented design of KTRL+F allows for various queries with different target answers for in-document searches.

Latency. Latency is a metric for assessing real-time applicability, therefore satisfying REQ 3. We measure in ms/Q (millisecond per query) which is widely used in retrieval systems to represent query inference speed (Khattab and Zaharia, 2020; Santhanam et al., 2022).

4 KTRL+F Dataset

We introduce a data construction pipeline to assemble essential components of KTRL+F: input document, query, corresponding targets, and external knowledge (Figure 2). Then we describe human verification procedures to ensure quality.

4.1 Dataset Construction Pipeline

Step 1. Select Real News Articles. To simulate real-world document scenarios, we randomly sample 100 English news articles from the publicly available C4 (Raffel et al., 2019) after preprocessing them based on their length and the number of entities. We utilize an entity linking API[2] to identify all entities within the article and extract external knowledge (i.e., Wikipedia) linked to the entities. Details of preprocessing and external knowledge are described in the Appendix A.

Step 2. Generate Pairs of (Query, Targets). Using the entities extracted from each input document (Step 1), we utilize LLAMA-2-Chat-70B (Touvron et al., 2023) to generate diverse queries and targets (prompt in Figure 5). We generate 10 questions for each input document. To satisfy the criteria of utilizing external knowledge (REQ 2), we provide only the extracted entities into the model, excluding the input document. This is done to remove the dependency on the document itself, as KTRL+F prioritizes queries that cannot be answered solely with the document and requires the integration of external knowledge.

Step 3-1. Target Filtering. To mitigate the potential problem of false positive and false negative in the generated targets by LLAMA-2-Chat-70B (Touvron et al., 2023), we implement an additional process inspired by Zhong et al. (2023). This process determines whether each entity is the answer to the query, leveraging external knowledge (prompt in Figure 6). Initially, we utilize GPT-3.5 (`gpt-3.5-turbo-0613`) (OpenAI, 2022) to identify entities judged as potential answer targets. Subsequently, GPT-4 (`gpt-4-0613`) (OpenAI, 2023) makes the final decision for entities where there is a disagreement between GPT-3.5 and the results of Step 2. Detailed statistics of the results by each model are available in the Appendix A.

Step 3-2. Query Filtering. Though we prioritize queries that require integrating external knowledge in Step 2, there are still many queries that do not meet REQ 2. To further reduce the number of such queries, we utilize a DeBERTaV3-large (He et al., 2023)[3], finetuned using the SQuAD 2.0 (Rajpurkar et al., 2018). We specifically exclude queries that the MRC model can answer solely based on the input document, leaving only suitable queries for REQ 2. Finally, 512 queries are collected out of the 1,000 queries generated in Step 2. See Appendix A for detailed scoring criteria of the MRC model.

[2] https://cloud.google.com/natural-language/docs/analyzing-entities

[3] https://huggingface.co/deepset/deberta-v3-large-squad2

Q1. Is it possible to answer using only the input doc?	
Need more external knowledge	74.3%
Don't need external knowledge	25.7%
% of answered targets	43.6%
Q2. Is it unnatural query?	
Natural Query	95.0%
Subjective Query	3.0%
etc.	2.0%
Q3. Reliability of Target determination	
kappa coefficient (κ)	0.627

Table 1: Human Verification Results

4.2 Dataset Analysis

Human verification setup. To assess the quality of the auto-generated dataset, we conduct human verification on a randomly selected subset of 104 queries, representing about 20% of the entire dataset. Eight annotators participated, with three assigned to evaluate each sample to minimize personal bias. Annotators are tasked with responding to three specific questions: two for query-side verification (*Q1* and *Q2*) and one for target-side verification (*Q3*).

The first question (*Q1*) assesses how well the generated query aligns with REQ 2. Annotators identify evidence for each target to answer the query, with the ideal response being annotators stating that evidence cannot be found in the input document for all targets. The second question (*Q2*) evaluates the naturalness of the generated query by choosing the type of unnatural query: *"Ambivalent or subjective expressions"*, *"Lack of factual basis"*, *"Logical errors"*, *"etc"*. The ideal response is for annotators to select *"None of these options"*, indicating a naturalness in the generated queries. The third question (*Q3*) focuses on evaluating the reliability of auto-generated targets. Annotators select the correct target for the query by referring to Wikipedia, mirroring the process in target filtering (Step 3-1) in the dataset construction pipeline. This establishes the reliability between the annotator's response and the dataset. Target-side verification is conducted on a distinct set of 104 samples from query-side verification. The user interface and detailed instructions for each question are presented in Figure 7.

Dataset quality and statistics. Since all samples are evaluated by three annotators, final human judgment is determined through majority voting. The inter-annotator reliability is detailed in Appendix B. For the first question, 74.3% of samples are considered unable to answer the target solely based on the input document. Of the remaining 25.7% of samples, only 43.6% of targets can be solved solely based on the input document. This indicates that our auto-generated dataset is suitable for evaluating KTRL+F requiring additional knowledge beyond the semantic information present in the input document. About the naturalness of query (Q2), 95% of samples are considered natural, while 3% are subjective. About 2% of the samples contain unnatural queries for other reasons, such as entities being directly mentioned in the query. For the third question, we find a kappa coefficient (Cohen, 1960) of $\kappa = 0.627$ between humans and the dataset. Following Landis and Koch (1977), this indicates *substantial agreement* between human judgment and the data construction pipeline. In total, the KTRL+F dataset comprises 512 queries for 98 input documents with an average of 4.2 mentions per query (Table 2). More examples of the KTRL+F dataset are available in Table 7.

	Avg.	Min.	Max.
Length of Input Document	1974	999	3254
Queries per Input Document	5.2	1	10
Answer Mentions per Query	4.2	1	30
Answer Entities per Query	1.8	1	8

Table 2: Statistics of KTRL+F Dataset

5 Knowledge-Augmented Phrase Retrieval

The challenge of KTRL+F is to effectively balance real-time applicability and high performance while utilizing efficient use of external knowledge. To meet the three requirements of KTRL+F, we propose Knowledge-Augmented Phrase Retrieval extending the phrase retrieval architecture of DensePhrases (Lee et al., 2021) within the setting of in-document search and enriching external knowledge about potential targets with external knowledge linking and knowledge aggregation modules as illustrated in Figure 3. Notably, our model doesn't require an additional training step.

5.1 External Knowledge Linking Module

The external knowledge linking module scans the target text, identifies entities that could be potential targets, and maps each of them to the relevant Wikipedia knowledge base. The module outputs

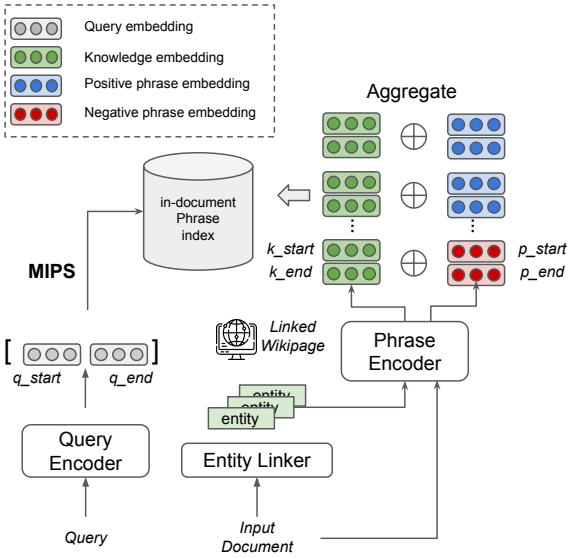


Figure 3: Overview of Knowledge-Augmented Phrase Retrieval.

a list of candidate targets along with the linked Wikipedia page for each target, serving as external knowledge about the targets. We use existing entity-liners to focus on building models that can integrate external knowledge. While there are various entity-linkers available, we choose to utilize a Wikifier API (Brank et al., 2017) as an entity linker for its ease of use.

5.2 Query and Phrase Encoder

The phrase and query encoder modules handle the encoding of the candidate phrase and the query, respectively. We utilize the pre-trained DensePhrases model (Lee et al., 2021) to extract phrase embeddings. For the query embedding, we extract the special token `[CLS]` from the output embeddings of the query encoder. We use two distinct query encoders to extract the start and end position embeddings for the query, following Lee et al. (2021). Subsequently, we concatenate the corresponding token embeddings, denoted as $[q_{start}; q_{end}] \in \mathbb{R}^{2d}$, to create a query embedding. Similarly, for the phrase encoder, we use concatenated token-level embeddings of the entity's boundary tokens (start and end token embeddings denoted as $[p_{start}, p_{end}]$) as the phrase embedding.

5.3 Knowledge Aggregation Module

To integrate external knowledge related to the entity, we employ the same phrase encoder used for extracting embeddings for candidate entities. Following the approach in Lee et al. (2023), we generate a knowledge embedding, denoted as $[k_{start}; k_{end}] \in \mathbb{R}^{2d}$, for the linked entity by concatenating the name of entity and its corresponding Wikipedia page (refer to Figure 8 for details). This effectively encodes relevant knowledge about the entity into its embedding. To combine external knowledge embedding with the entity embedding and create an in-document phrase index, we use a straightforward element-wise addition operation. This demonstrates promising results in our experiments enabling the system to capture the contextual knowledge for more accurate and comprehensive search and retrieval within the document without requiring further tuning. Through the Maximum Inner Product Search (MIPS) operation, Knowledge-Augmented Phrase Retrieval can identify all matching targets in real time.

6 Experiments

6.1 Setup

When selecting baselines, our primary focus lies in evaluating the effectiveness of various representative options in addressing KTRL+F. We categorize potential baseline types into generative, extractive, and retrieval (ours) models.

Generative baselines solve KTRL+F as a text generation problem, where the model takes instructions, a input text, and a query as input and sequentially produces matching targets (see Appendix C). The parametric space of Large Language Models (LLM) serves as an implicit source of general knowledge under the assumption that LLMs can serve as a closed-book model, as discussed by Raffel et al. (2019); Roberts et al. (2020); Brown et al. (2020); De Cao et al. (2020); Yu et al. (2023). To explore the knowledge within the parametric space, we utilize various LLM models, such as the LLM API versions GPT-3.5 (OpenAI, 2022) and GPT-4 (OpenAI, 2023), as well as open-source models like LLAMA-2 (Touvron et al., 2023) and VICUNA v1.5 (Chiang et al., 2023), ranging in size from 7B to 13B. We additionally post-process generated outputs of models to only extract targets for evaluation.

Moreover, we observe that Retrieval Augmented Generation (RAG) baselines, which merely retrieve and enhance information from the query side, performs worse than naive LLM approaches. The unique characteristics of KTRL+F require grounding information from both the query and target

Type	Model	Speed	Performance			
		Latency (ms/Q) (↓)	List EM (↑)	(R) List EM (↑)	List Overlap (↑)	(R) List Overlap (↑)
Generative	GPT-3.5	-	30.346	8.284	**41.929**	19.446
	GPT-4	-	**30.457**	7.452	37.402	12.898
	LLAMA-2-Chat-7B	2359	28.529	**8.947**	40.546	20.008
	LLAMA-2-Chat-13B	3176	28.846	8.024	37.098	14.367
	VICUNA-7B-v1.5	1951	17.831	3.694	31.216	12.532
	VICUNA-13B-v1.5	2420	24.490	6.977	39.278	20.401
Extractive	SequenceTagger	26	7.239	0.612	8.614	1.211
Retrieval	Ours (w/ Wikifier)	**15**	23.152	7.091	40.718	**23.107**
	Ours (w/ Gold)	14	46.170	22.426	53.689	32.285

Table 3: Speed and performance evaluation results for KTRL+F dataset. Note that API-based models (GPT-3.5 and GPT-4) are excluded from speed evaluation. Robustness scores are noted with (R) with corresponding metric. Ours denotes Knowledge-Augmented Phrase Retrieval, and the best results excluding Ours (w/ Gold) are in bold, while second-best ones are underlined.

Entity Linker	Model	List EM (↑)	(R)List EM (↑)	List Overlap (↑)	(R)List Overlap (↑)
Gold (GCP API)	**Ours**	46.170	22.426	53.689	**32.285**
	- External	34.582	14.178	43.758	26.406
	- Internal	**47.345**	**23.097**	**54.308**	30.599
Wikifier	**Ours**	**23.152**	7.091	**40.718**	**23.107**
	- External	15.620	4.742	31.805	18.823
	- Internal	22.851	**7.773**	39.391	20.812

Table 4: Ablation study on the impact of existence and quality of external knowledge. We measure the performance when using different entity linkers (Gold w/ GCP API, Wikifier API). We further evaluate the impact of contextual phrase embedding (Internal) and external embedding (External) by removing the related part.

text sides, presenting a distinct challenge. Consequently, existing methods in the RAG models, which focus solely on retrieving knowledge from the query side,fail to adequately address this challenge. For detailed results and analysis of RAG baselines, please refer to Appendix D.

Extractive baseline is similar to extraction-based model for Machine Reading Comprehension task. This approach uses the internal knowledge within the target text to directly locate the answer spans. In order to find all relevant spans in the target text, we follow the previous works (Segal et al., 2020; Li et al., 2022) that helps identify multiple entities. We utilize a BERT based sequence tagging model which is fine-tuned using MultiSpanQA (Li et al., 2022) dataset, denoted as SequenceTagger.

6.2 Results

Lower latency means faster time to find targets[4], and among various metrics, the List Overlap score can be indicative of general performance[5]. Note that all models in the experiment are evaluated in a zero-shot manner.

Generative and extractive baselines show difficulties in balancing real-time applicability and performance as Table 3. GPT-3.5 excels in List Overlap scores, leveraging its parametric knowledge effectively. Interestingly, expanding model capacity doesn't consistently enhance performance unlike increasing latency. Upon close examination of LLAMA-2 models, we can find possible reasons: smaller models generate more targets (avg. 3.347 for 7B, avg. 2.324 for 13B), leading to lower precision but higher recall, ultimately contributing to improved performance in List Overlap. The generative nature of these models introduces complexities including challenges such as hallucination and difficulties in effective restriction of generated output (see examples in Table 16). Conversely, the SequenceTagger, an extractive baseline, falls short in KTRL+F. Its inability to utilize external knowledge highlights the importance of incorporating such knowledge beyond the input document for successful KTRL+F resolution. For a comprehensive baseline understanding, prediction example for

[4]Speed measurements use an A6000 GPU on a server with two AMD EPYC 7513 CPUs, each with 32 physical cores.

[5]We report the micro-averaged F1 scores. Detailed Precision and Recall scores are available in Table 8.

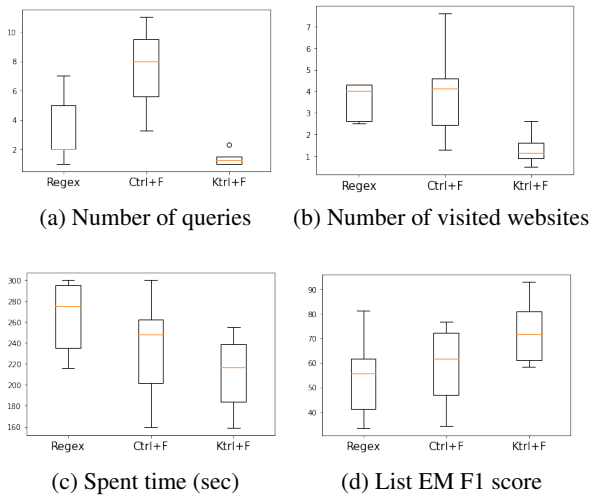
each model is available in Appendix F and additional experiments are reported in Appendix G.

Knowledge-Augmented Phrase Retrieval demonstrates a balance between latency and achieving overall performance. Incorporating knowledge embedding into the phrase retrieval process, our model (Ours w/ Wikifier) demonstrates competitive performance in List Overlap metrics, despite having a significantly smaller model capacity (330M, only 5% of the smallest generative baseline) than other generative baselines. When provided with gold entity linking information used in the dataset construction pipeline, our model achieves the best performance (Ours w/ Gold). To compare with other baselines, we threshold the prediction results from top 4 according to the data distribution [6]. Beyond performance, the retrieval-based design of our model is suitable for real-time applicability, exhibiting smaller latency than other baselines. While our model demands extra time for the initial indexing of long input documents into searchable format, taking 2.863 and 0.955 seconds for our models with Wikifier and Gold respectively, the subsequent querying of the indexed text introduces real-time latency. This shows a significant advantage compared to generative baselines, even when utilizing the LLM acceleration methods (see Appendix J).

6.3 Ablation Study

We evaluate the importance of the knowledge aggregation design in our model. Our model utilizes an in-document phrase index by adding knowledge embedding from Wikipedia and phrase embedding from the input document. In Table 4, (-External) excludes external knowledge embedding, and (-Internal) removes phrase embedding. Results indicate a notable performance drop with (-External) when both entity linkers are used. When phrase embedding is removed (-Internal), the model with the Gold entity linker performs better overall, while the model with Wikifier shows lower results compared to using both embeddings. However, robustness of List Overlap scores consistently remains higher than when partial components are removed, emphasizing the vital role of internal knowledge in constructing a resilient embedding, particularly when external information quality is suboptimal.

[6]To provide a comprehensive understanding of the model, we additionally report MAP metrics in Table 9 of Appendix E.



(a) Number of queries (b) Number of visited websites

(c) Spent time (sec) (d) List EM F1 score

Figure 4: A comparison of in-document search systems. Ktrl+F plugin outperforms other systems overall.

7 User Study

To verify whether solving KTRL+F can enhance search experience of users in the real web environments, we build Chrome extension plugin (KTRL+F plugin) built on our model.

7.1 Setup

Each user is assigned to use only a specific system per example among KTRL+F plugin, Ctrl+F, and Regular expression to help them find all targets that match given search intent from a given website. Criteria for evaluation are shown in Figure 4. Further details for the user study are provided in Appendix H.

7.2 Findings

For a comprehensive comparison of the usefulness and efficiency of the KTRL+F plugin with other in-document search systems, we present the results of the conducted user study in Figure 4.

Less search time with KTRL+F plugin. As depicted in Figure 4 (c), the KTRL+F plugin exhibits the shortest time when searching for targets. This efficiency stems from its capacity to identify multiple semantic targets in a single query, minimizing the need for additional searches to validate results. While regular expressions can similarly search for multiple targets simultaneously, the process involves complex creation and often difficult debugging, as exemplified in Figure 15 of Appendix H.

Fewer queries to find targets. Figure 4 (a) illustrates the average number of queries used to find

answers. Regular expressions and Ctrl+F rely on user-generated candidate lexical prefixes to find answers. Transforming search intent into the format supported by these systems increases query usage. While Ctrl+F allows swift query verification, users struggle to predict which keywords will appear in unknown text before reading it entirely. Regular expressions can consolidate multiple simple searches into one, but dynamically crafting complex expressions is challenging and debugging erroneous code compounds the complexity.

Fewer visits for extra sources. The ability to extend external knowledge beyond the current web page of KTRL+F plugin alleviates the need to consult additional sources to verify results, as shown in Figure 4 (b). Additionally, users often overlook variations when using manual lexical matching systems. For example, in the query "List all football teams from the web page," users might overlook variations such as Liverpool FC's nickname "The Reds." The ability to handle such subtle changes of KTRL+F plugin contributes to improved performance as Figure 4 (d).

8 Conclusion & Future Work

In this paper, we introduce KTRL+F, a knowledge-augmented in-document search that requires identifying all semantic targets with a single natural query in real-time. KTRL+F tackles unique challenge for in-document search that requires capturing targets containing additional information beyond the input document by utilizing external knowledge while balancing speed and performance. We highlight limitations in existing models, such as hallucinations, high latency, or difficulties to incorporate external knowledge. And show that our Knowledge-Augmented Phrase Retrieval, simple extension of phrase retrieval architecture can be a robust model for KTRL+F. Moreover, the study demonstrates that even our straightforward model, with seamless access to external knowledge during in-document searches, significantly enhances the user search experience.

Future work could extend KTRL+F to reflect updated knowledge, such as news, or domain-specific knowledge bases, such as the medical domain, which cannot be easily handled by large language models alone (Ram et al., 2023; Peng et al., 2023; Kaddour et al., 2023). The scalability and practicality of KTRL+F will open up opportunities for various advancements in the field of information retrieval and knowledge augmentation.

Limitations

The system design for KTRL+F can incorporate various forms of external knowledge, not limited to the Wikipedia page associated with the entity. It can also identify a wide range of target spans within the target text, including dates and numbers, without being restricted to entities. However, the primary focus of this paper revolves around addressing KTRL+F, specifically emphasizing entities as the primary search targets. By narrowing our focus to entities, we make effective use of entity linking information as external knowledge. Furthermore, due to the inherent nature of retrieval systems, our Knowledge-Augmented Phrase Retrieval model requires an extra indexing stage whenever a change in the input document, which requires additional time to use. Also it relies on thresholding to truncate predicted results, which we employ top-k results based on the data distribution in our experiment. Exploring more efficient methods for enhancing external knowledge while reducing the time needed for the indexing stage is a potential avenue.

Acknowledgements

This work was partly supported by Samsung Research grant (2021, Multi-grained Passage Embedding via Cross-to-Bi Encoder Distillation, 80%) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub, 20%).

References

Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew R Gormley. 2023. Unlimiformer: Long-range transformers with unlimited length input. *arXiv*.

Janez Brank, Gregor Leban, and Marko Grobelnik. 2017. Annotating documents with relevant wikipedia concepts. In *SiKDD*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS*.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024.

Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *EMNLP*.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *NAACL*.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *arXiv*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. 2020. Iirc: A dataset of incomplete information reading comprehension questions. In *EMNLP*.

Thibault Févry, Livio Baldini Soares, Nicholas Fitzgerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. In *EMNLP*.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*.

Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. 2024. Break the sequential dependency of llm inference using lookahead decoding. *arXiv*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. In *ICLR*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *EACL*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *arXiv*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *SIGIR*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *TACL*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *SIGOPSSIGOPS*.

J Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*.

Hyunji Lee, Jaeyoung Kim, Hoyeon Chang, Hanseok Oh, Sohee Yang, Vladimir Karpukhin, Yi Lu, and Minjoon Seo. 2023. Nonparametric decoding for generative retrieval. In *Findings of ACL*.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning dense representations of phrases at scale. In *ACL*.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. Sensebert: Driving some sense into bert. In *ACL*.

Haonan Li, Martin Tomko, Maria Vasardani, and Timothy Baldwin. 2022. Multispanqa: A dataset for multi-span question answering. In *ACL*.

Sheng-Chieh Lin, Minghan Li, and Jimmy Lin. 2022. Aggretriever: A simple approach to aggregate textual representations for robust dense passage retrieval. *TACL*.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

OpenAI. 2023. Gpt-4 technical report.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv*.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *EMNLP-IJCNLP*.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-bert: Efficient-yet-effective entity embeddings for bert. In *Findings of EMNLP*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Vatsal Raina, Nora Kassner, Kashyap Popat, Patrick Lewis, Nicola Cancedda, and Louis Martin. 2023. Erate: Efficient retrieval augmented text embeddings. In *First Workshop on Insights from Negative Results in NLP*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *ACL*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *TACL*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *EMNLP*.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *NAACL*.

Elad Segal, Avia Efrat, Mor Shoham, Amir Globerson, and Jonathan Berant. 2020. A simple and effective model for answering multi-span questions. In *EMNLP*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuan-Jing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-adapter: Infusing knowledge into pre-trained models with adapters. In *Findings of ACL-IJCNLP*.

Ledell Yu Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval. In *EMNLP*.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *arXiv*.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *ICLR*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In *ACL*.

Victor Zhong, Weijia Shi, Wen-tau Yih, and Luke Zettlemoyer. 2023. Romqa: A benchmark for robust, multi-evidence, multi-answer question answering. In *Findings of EMNLP*.

Ming Zhu, Aman Ahuja, Da-Cheng Juan, Wei Wei, and Chandan K. Reddy. 2020. Question answering with long multiple-span answers. In *EMNLP*.

A Details for Dataset Construction Pipeline

Step 1. Select Real News Articles. The preprocessing of articles involves two criteria. First, 6,936 articles are collected from the 13,863 articles in the C4 realnewslike validation set, with lengths ranging from 991 to 3,298, covering the lower to upper quartiles to remove abnormal articles. Then, to ensure diversity of questions and quality of documents, we collect 3,910 articles with 4 to 11 entities, covering the lower to upper quartiles.

We consider Wikipedia through October 31, 2023 as an external knowledge source. The acquisition of external knowledge for targets is equated to utilizing the corresponding Wiki page linked to a particular entity. (Wu et al., 2019).

Step 3.1. Target Filtering. In this step, given a (query, entity, external knowledge) triple, we follow Zhong et al. (2023) to derive whether an entity is an answer to a query or not. We utilize the first 10 sentences from the Wikipedia article as an external knowledge, which covers more than 99% of the total sample within 4,096 tokens of GPT-3.5. GPT-3.5 processes a total of 7,060 triple samples, and the final judgment is made by GPT-4 on 1,226 samples that show different results from the target generated by LLAMA-2 in Step 2. On average, 1.6 entities disagreed per query, which is an average of 22% of the candidate entities per query. After the final judgment, queries with all targets determined to be false are discarded. As a result, 816 queries remained out of the total 1,000 queries generated by Step 2, and the average number of entities in a target increased slightly from 1.4 to 1.9.

Step 3.2. Query Filtering. In this step, we exclude a query if the MRC model answers any of the target entities. The MRC model is considered correct when it scores over 0.9 in F1 score, following the human performance described in Rajpurkar et al. (2018). As a result, 512 queries were collected from the 816 queries derived in Step 3-1.

B Inter-Annotator Reliability of Human Verification

Eight annotators, all of whom are computer science majors proficient in English participated Human verification. To assess the inter-annotator reliability among the three annotators, we utilize Fleiss' kappa value (Fleiss, 1971), a metric used to evaluate the agreement between multiple annotators in assigning categorical ratings. We follow the interpretation of kappa value by Landis and Koch (1977): < 0 indicates *poor aggreement*; 0.01-0.20 indicates *slight agreement*; 0.21–0.40 indicates *fair agreement*; 0.41–0.60 indicates *moderate agreement*; 0.61–0.80 indicates *substantial agreement*; and 0.81–1.00 indicates *almost perfect agreement*.

The first and second questions, classified as query-side verifications, scored kappa values of 0.552 and 0.4458 respectively, indicating *moderate agreement* among the three annotators. In contrast, the third question scored 0.7193, indicating *substantial agreement*. The nature of query-side verification, which relies on subjective evaluations, tends to result in lower inter-annotator reliability compared to target-side verification. The latter involves objective fact-checking with reference to Wikipedia, leading to higher agreement among annotators.

C Implementation Details for Baselines

Generative baselines. To convert KTRL+F as generation problem, we use following instructions for generative models and then post-process the output text to only utilize the answer part. We use temparture 0.5, max new token 512.

```
Find all mentions from the article
below that correspond to the query.
Only generate mentions with comma
separate.

Article: {Input Document}
Query: {Query}
Mentions:
```

Extractive baseline. We solve KTRL+F using sequence tagging model following (Li et al., 2022). It can be regarded as a model without utilizing external knowledge. We reproduce the model trained on MultiSpanQA (Li et al., 2022) for 3 epochs.

D Analysis of RAG baselines

We append the top-5 retrieved passages using DensePhrases (Lee et al., 2021) as a retriever to the LLM input. Here is the prompt we utilized for the RAG experiment.

```
Find all references to your query
in the ARTICLE below, referring to
the external evidence provided.

- Generates only matching pairs of
  mentions from the ARTICLE, separated
  by commas. Just generate answers!
  This is IMPORTANT.
```

Model	List EM (↑)	(R) List EM (↑)	List Overlap (↑)	(R) List Overlap (↑)
RAG-GPT-3.5	8.338	2.233	27.404	13.573
RAG-GPT-4	**28.279**	**8.457**	**42.791**	**20.646**
RAG-LLAMA-2-Chat-7B	7.987	2.361	28.465	16.469
RAG-LLAMA-2-Chat-13B	9.140	2.262	26.949	12.894
RAG-VICUNA-7B-v1.5	4.468	0.770	24.685	12.156
RAG-VICUNA-13B-v1.5	5.773	1.163	28.745	16.860

Table 5: Results for RAG baselines. We utilize DensePhrases as a retriever and augment top 5 retrieved passages from the Wikipedia dump provided by the authors to the LLM input.

```
- Do NOT extract mentions from the
  EVIDENCE.
- If a same mention appears multiple
  time, generate every mentions.
- Please do not generate any other
  opening, closing, and explanations.
  Just generate the set of scenarios!

#Evidence: {top-k paragraphs}
#Article: {target_text}
#Query:{query}
#Mentions:
```

Despite explicitly providing additional information, incorporating retrieval information into the LLM input diminishes performance compared to a straightforward LLM approach. Notably, performance declines significantly across all models except for GPT-4, as demonstrated in Table 5. Upon manual analysis, we observe that the retrieval system adequately retrieves paragraphs related to the query in general. However, two types of errors are identified: 1) failure to retrieve relevant targets for the target text during the retrieval stage, and 2) failure to ground instructions that not only extract information solely from the target text (the article) but also extract answers from the retrieved evidence during the generation stage.

For instance, when using 'Social media platforms' as the retrieval query, one of the retrieval results includes descriptions about various platforms such as Facebook, MySpace, YouTube, and blogs. However, in the corresponding target text to be skimmed, there are no relevant sections within the retrieved paragraphs, and the only target we can match from the provided target text is 'Twitter'. In this scenario, the retrieved paragraphs can serve as distractors for the generative model, making it challenging to extract information solely from the target text, as indicated in the experimental table. We emphasize that the unique characteristics of our datasets in KTRL+F demand grounding information from both the query-side and target text side, presenting a distinct challenge.

E Further Analysis of Retrieval Approach for KTRL+F

Determining a proper threshold for retrieval is challenging, especially when the number of targets varies. Therefore, we additionally measure the Mean Average Precision (MAP), which calculates the mean value per query Q of the Area under the Curve (AUC) of the precision-recall graph in Table 9. This metric provides a comprehensive measure of the system's ability to quantify the overall effectiveness.

F Prediction Examples

Table 12 shows the results of various approaches on same query and input document for qualitative analysis.

G Baseline Analysis from Different Perspectives

For a comprehensive baseline understanding, we additionally present set-base scores which doesn't require recognizing every target occurrences in Table 10. We can see the Set Overlap score gets a higher result than List Overlap overall, and especially generative models show major performance gain in Overlap score when using Set score, which shows finding all matching target is hard for generative models. Given that our model leverages entity linking information to identify targets from a restricted pool of candidates, we conduct an additional experiment by supplying additional information about potential targets for generative models (refer to Table 11). When adding extra information about potential targets for generative models, it proves to enhance the overall performance of generative models. Notably, in the case of LLM-API (GPT-4, GPT-3.5), it even outperforms our

model with gold-standard information. However, it's important to note that enhancing information for generative models comes with increased costs and slower latency, making it impractical for real-time applicability.

H Details for User Study

We compare existing in-document search systems in Table 13, considering criteria such as matching type, the system's ability to search multiple targets, its search intention, and its capacity to augment external knowledge. Additionally, Table 15 includes examples of queries users employ with different in-document search systems to find the same targets.

We recruit six participants from the computer science field, each solving 10 examples from designated websites. For each example, we assign two individuals per tool to enable us to collect responses using three different tools (Ctrl+F, Regex, KTRL+F plugin) for each example. To present users with challenging search goals that require identifying multiple target variants within a document, we believe that leveraging the dedicated KTRL+F dataset tailored for this purpose was a natural choice. Thus, we select all examples linked to our Ktrl+F dataset. The participants manually annotate the targets in the PDFs using the respective system. For in-depth analysis, all experiments are conducted on-site and we record the screens of participants throughout the experiment to capture the entire search process. Instructions given to the participants are as follows:

```
- Click Web Page URL
- Find all candidate spans in the
  Web page which meets noted search
  intention.
- You can utilize Answer information
  when you are using Ctrl+F & Regex
- NOTE: use only specified used system
  per example
- All extraction should be highlighted
  manually in the linked PDF URL
- You only have to fill spent time per
  example manually in this sheet
  (max 5min per example)

(FYI, You can search multiple targets
 using Regex in this format :
 \b(?:SAN JOSE|Calif|Anaheim)\b)
```

I Details for List Overlap F1 Metric

The List Overlap F1 score follows the definition of span overlap as outlined in MultispanQA (Li et al., 2022). Equation 1 calculates the partial retrieved and relevant scores for each pair (p_i, g_i) by determining the length of the longest common substring (LCS) and dividing it by the length of the respective spans.

$$\begin{aligned} s_{ij}^{ret} &= len(LCS(p_i, g_i))/len(p_i) \\ s_{ij}^{rel} &= len(LCS(p_i, g_i))/len(g_i) \end{aligned} \tag{1}$$

Different from set-based F1, List Overlap identify all occurrences. When there are n predicted occurrences and m target occurrences for a question, all metrics are defined as below.

$$\begin{aligned} Precision &= \frac{1}{n}\sum_{i=1}^{n} \max_{j\in[1,m]}(s_{ij}^{ret}) \\ Recall &= \frac{1}{m}\sum_{j=1}^{m} \max_{i\in[1,n]}(s_{ij}^{rel}) \\ F1 &= \frac{2 \times Precision \times Recall}{Precision + Recall} \end{aligned} \tag{2}$$

J Comparison with LLM Acceleration Methods

In the Table 3, the generative baselines show poor latency relative to its performance. We compare how much the generative method can compensate for latency through acceleration methods, including algorithmic acceleration methods such as Lookahead Decoding (Fu et al., 2024) and Medusa (Cai et al., 2024), as well as hardware-level acceleration such as vLLM (Kwon et al., 2023). The Medusa (Cai et al., 2024) shows nearly 2x speedup, but still lagging behind retrieval methods. However, even without any low-level optimizations, our retrieval-based method is still more efficient than generative approaches. Considering real-time latency as a key requirement for KTRL+F, exploring generative approaches in this problem holds promise for future research.

Model	Latency (ms/Q) (↓)
Vicuna-7B-v1.5	1951
+ Lookahead	1520
+ vLLM	1277
+ Medusa	1012
Vicuna-13B-v1.5	2420
+ Lookahead	2046
+ vLLM	1749
+ Medusa	1280
Ours	**15**

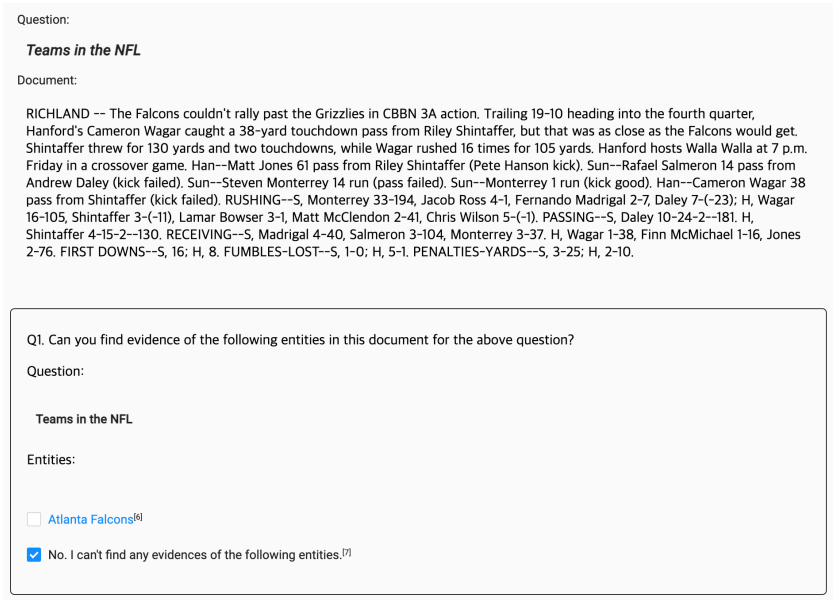
Table 6: Comparison of latency on Vicuna with acceleration methods.

```
You are an expert of query generation for entity search.

You must follow this requirements.
Requirements:
- Your task is to generate queries that retrieve entities in a given list.
- The generated query must be able to list the multiple entities.
- The answers must be countable.
- The answers have to be entities.
- Make sure your questions are unambiguous and based on facts rather than temporal information.
- Do not specify the number in a query.
- Do not start with 'What' in a query.
- Do not start with 'Which' in a query.
- Do not include an expression in your query that tells it to find from a given list.

The example is as below.

Generate 4 queries from the following list and extract subset list.
Candidate List:
[Apple, Microsoft, Samsung Electronics, Alphabet, AT&T, Amazon, Verizon Communications, China Mobile, Walt
    Disney, Facebook, Alibaba, Intel, Softbank, IBM, Tencent Holdings, Nippon Telegraph & Tel, Cisco
    Systems, Oracle, Deutsche Telkom, Taiwan Semiconductor, KDDi , HP, Legend Holding, Lenovo Group, ebay]

Query:
1. IT companies in Computer Hardware industry
=> Apple, HP, Legend Holding, Lenovo Group
2. Find all IT companies that have software as main business.
=> Microsoft, Oracle
3. Companies that is known for retail service
=> Amazon, Alibaba, ebay
4. Name all IT companies that have license in USA
=> Apple, Microsoft, Alphabet, AT&T, Amazon, Verizon Communications, Walt Disney, Facebook, Intel, IBM,
    Cisco Systems, Oracle, HP, ebay
```

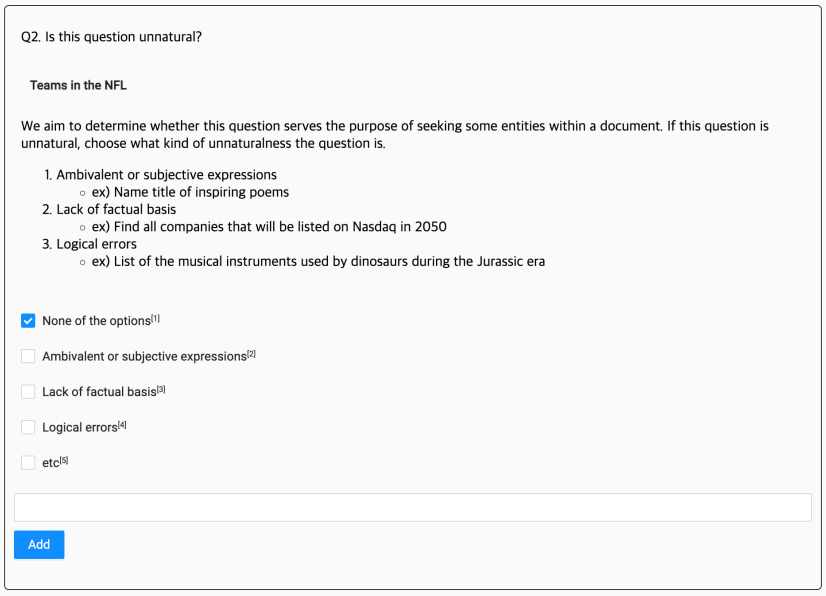
Figure 5: Prompt for generating queries and targets

```
You are a QA system to identify the given entity is the answer.
The inputs are entity, query and evidence.

You must follow this requirements.
Requirements:
- Output have to be either 'true' or 'false'
- Do not say anything except 'true' or 'false'

The example is as below.

Entity: Google
Query: Find all IT companies in Computer industry
Evidence: Google LLC is an American multinational technology company focusing on artificial intelligence,
    online advertising, search engine technology, cloud computing, computer software, quantum computing,
    e-commerce, and consumer electronics. It has often been considered "the most powerful company in the
    world" and as one of the world's most valuable brands due to its market dominance, data collection,
    and technological advantages in the field of artificial intelligence. Its parent company Alphabet is
    often considered one of the Big Five American information technology companies, alongside Amazon,
    Apple, Meta, and Microsoft.
Output: true

Entity: Samsung
Query: Find all companies in United States
Evidence: Samsung Group, or simply Samsung, is a South Korean multinational manufacturing conglomerate
    headquartered in Samsung Town, Seoul, South Korea. It comprises numerous affiliated businesses, most
    of them united under the Samsung brand, and is the largest South Korean chaebol (business
    conglomerate). As of 2020, Samsung has the eighth highest global brand value.
Output: false
```

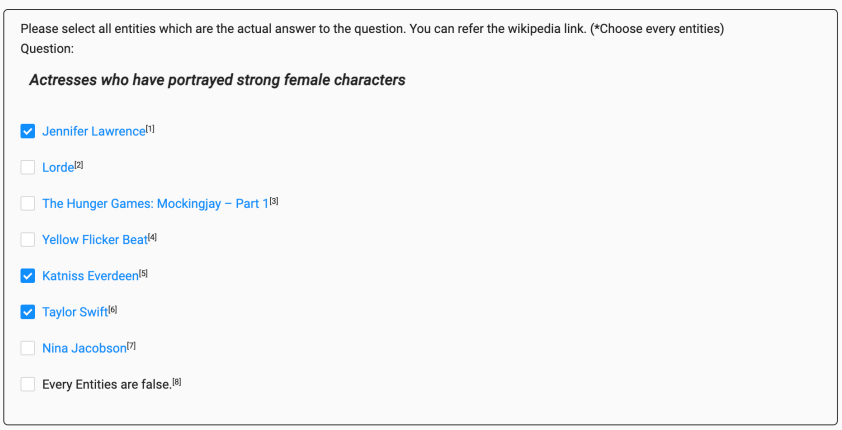
Figure 6: Prompt for target filtering



(a) The Q1 requests the identification of evidence for each target to evaluate whether the query satisfies REQ 2.



(b) The Q2 requests the selection of options to evaluate the naturalness of the query.



(c) The Q3 requests the selection of targets to evaluate the reliability of target determination.

Figure 7: User Interface for Human Verification.

[Query] ***Entities that are known for their cookie products***

[Input Document]

Nabisco threatens to sue a Canadian man who registered " oreos.com " for his home page with adult links. " Oreos.com " sat quietly on the Net for more than a year–however, it wasn't a hub to debate whether the cookie's crunchy chocolate outside is better than its creamy filling. On the contrary, until today, " Oreos.com " was an Ontario man's personal Web page featuring links to some adult entertainment sites. While this may have been a treat for some, it is not exactly the one most people affiliate with Nabisco 's famous sandwich cookie. And so it was that Paul Figueiredo found himself in a legal dispute with one of the biggest food companies in the United States and Canada. Nabisco threatened a lawsuit if he didn't surrender the domain name by noon today. ... At first, " Oreos.com " was registered to be the site for the Ontario Real Estate Online Services, for advertising homes on the market, Figueiredo says. But his business idea never took off. He had already spent $100 to register the site name, so he turned it into a home page. When he got the letter from Nabisco 's lawyers earlier this month, he knew the site's days were numbered. ... He tried unsuccessfully to cut a deal that would have allowed him to point people to Nabisco 's official site, if he took the adult links off the front page. But because the company markets its sweets to kids, Nabisco wouldn't go for it, he said. "If it was my kid, I wouldn't want them to see adult banners when they type in ' Oreos ,'" he admitted. ... And most won't take "no" for an answer. "He has faxed back the letter agreeing to cease all use of the Oreos trademark and domain name," Jonathan Colombo, an attorney for Nabisco , said today.

[Query] ***Companies that offer cloud computing services***

[Input Document]

Razer's latest eGPU cabinet gets LEDs and a bigger PSU , plus a ton more ports than before. Alienware's redesigned powerhouse laptop promises the Holy Grail of gaming laptop features. It's big, fast, beautiful, and even upgradable. Google 's shown it can kill off a product when it no longer deserves to live. We know a few more products that are ready to die, if only Google could help. We go hands-on with HP's Reverb Consumer Edition, whose astounding resolution is well deserving of this exclamation mark! Here's what you need to know about Maxon's new Cinebench R20 benchmark, and how to use it to test your computer. Acer's Predator Helios 300 is currently the bestselling gaming laptop on Amazon . With an 8th-gen Core i7, GeForce GTX 1060, and 144Hz screen, it's easy to see why. We delve into those and other details. Give the ThinkPad six cores and a GeForce GPU and you get the Lenovo ThinkPad Extreme X1, a 15-inch laptop that's large and in charge. Lenovo's newest mainstream IdeaPad laptops give you a choice between Ryzen and RX Vega, or Core i7 and a mystery GeForce MX graphics.

[Query] ***Cities in Wisconsin***

[Input Document]

Workers wear double-lined suits, and the floor is heated to prevent permafrost. In one of the coldest workplaces on earth, in New Berlin , employees wear heated boots with a 2-inch-thick sole. Inside their work area — two freezers totaling 12,000 square feet — it's nearly 70 below zero, colder than most winter days in Siberia. ... Cultures are stored at minus-67 degrees until they're shipped, frozen, to food companies that thaw them and put them to work making products. The company also makes probiotic bacteria strains for health care companies around the world. "We develop and produce cultures, enzymes, probiotics and natural colors for a rich variety of foods, confectionery, beverages, dietary supplements and even animal feed and plant protection," the company says. More than 1 billion people a day consume products containing the company's natural ingredients, the Chr Hansen website says. The company has more than 3,000 employees, in about 30 countries, including about 300 in New Berlin and the Milwaukee area. It was founded by a Danish pharmacist in 1874 and has been in the Milwaukee area since the late 1920s. "We've been pretty fortunate in the people we've been able to recruit and retain," Graham said.

[Query] ***Sports teams in the state of Georgia***

[Input Document]

RICHLAND – The Falcons couldn't rally past the Grizzlies in CBBN 3A action. Trailing 19-10 heading into the fourth quarter, Hanford's Cameron Wagar caught a 38-yard touchdown pass from Riley Shintaffer, but that was as close as the Falcons would get. Shintaffer threw for 130 yards and two touchdowns, while Wagar rushed 16 times for 105 yards. Hanford hosts Walla Walla at 7 p.m. Friday in a crossover game. Han–Matt Jones 61 pass from Riley Shintaffer (Pete Hanson kick). Sun–Rafael Salmeron 14 pass from Andrew Daley (kick failed). Sun–Steven Monterrey 14 run (pass failed). Sun–Monterrey 1 run (kick good). Han–Cameron Wagar 38 pass from Shintaffer (kick failed). ...

Table 7: Example of KTRL+F evaluation dataset. The highlights indicate target mentions and link to the Wikipedia page. For example, in the fourth sample, "Falcons" links to the Wikipedia page for "Atlanta Falcons".

Model	List EM			List Overlap		
	Precision	Recall	F1	Precision	Recall	F1
GPT-3.5	39.2	33.0	30.3	54.2	49.3	41.9
GPT-4	39.0	31.8	30.4	49.0	41.6	37.4
LLAMA-2-Chat-7B	28.5	35.9	28.5	46.6	49.9	40.5
LLAMA-2-Chat-13B	37.5	29.5	28.8	50.1	40.9	37.0
VICUNA-7B-v1.5	24.3	21.9	17.8	39.2	44.9	31.2
VICUNA-13B-v1.5	29.1	36.1	24.4	43.5	56.1	39.2
SequenceTagger	12.6	6.3	7.23	18.8	7.6	8.6
Ours (w/ Wikifier)	23.7	33.5	23.1	48.3	47.4	40.7
Ours (w/ Gold)	47.7	63.6	46.1	61.2	64.6	53.6

Table 8: Detailed performance evaluation including Precision and Recall for KTRL+F dataset.

Model	Indexing time (Sec) (↓)	ms/Q (↓)	MAP(@IoU0.5) (↑)	(R)MAP(@IoU0.5) (↑)
Ours w/ Wikifier	3.555	14	0.464	0.209
w/o INT	3.027	14	0.494	0.220
w/o EXT	3.145	14	0.335	0.153
Ours w/ Gold	0.955	14	0.716	0.380
w/o INT	0.912	14	0.776	0.408
w/o EXT	0.799	14	0.508	0.213

Table 9: MAP metric for retrieval approach. The result shows the effectiveness of phrase retrieval architecture. When using MAP as a metric, it reflect retrieved ranks of results and ours show slightly performance drop than ours w/o internal knowledge.

Model	List EM (↑)	Set EM (↑)	List Overlap (↑)	Set Overlap (↑)
GPT-4	30.457	36.422	37.402	51.071
GPT-3.5	30.346	36.668	41.929	56.334
LLAMA-2-Chat-7B	28.529	34.235	40.546	52.843
LLAMA-2-Chat-13B	28.846	35.206	37.098	51.672
VICUNA-7B-v1.5	17.831	22.265	31.216	42.460
VICUNA-13B-v1.5	24.490	29.223	39.278	49.449
SequenceTagger	7.239	9.041	8.614	15.648
Knowledge-Augmented Phrase Retrieval (w/ Wikifier)	23.152	24.793	40.718	46.841
Knowledge-Augmented Phrase Retrieval (w/ Gold)	46.170	50.254	53.689	63.230

Table 10: We additionally report Set-based scores with our List-based scores, which doesn't necessitate recognizing every target occurrences.

Model	List EM (↑)	(R) List EM (↑)	List Overlap (↑)	(R) List Overlap (↑)
GPT-4 (w/ Gold)	**52.937**	**22.479**	55.765	25.183
GPT-3.5 (w/ Gold)	44.697	22.048	**56.615**	**35.874**
LLAMA-2-Chat-7B (w/ Gold)	40.225	17.738	50.466	30.140
LLAMA-2-Chat-13B (w/ Gold)	45.674	19.329	50.172	23.291
VICUNA-7B-v1.5 (w/ Gold)	27.374	8.651	41.466	21.611
VICUNA-13B-v1.5 (w/ Gold)	39.898	17.065	54.695	33.814
Knowledge-Augmented Phrase Retrieval (w/ Gold)	46.170	22.426	53.689	32.285

Table 11: Results for when generative models use candidate entities from input document as additional input for instruction (denoted as w/ Gold). We evaluate the results by giving gold entity linking information version.

[Query] ***Social network platform of China***

[Input Document]

It is a highly competitive market with many local competitors who already understand the shopping habits of the Chinese, which are very different to those of consumers in the Western world. Chinese platforms such as Taobao and Tmall dominate the shopping world … successfully. BAIDU DOMINATES ONLINE Currently, there are an estimated 900 million internet users across China, with most users spending 1.5 hours a day just browsing. Baidu is the most popular search engine across China. Think of it as 'the Google of China'. … time. Baidu also brings the … social media app across China, it is imperative that your company becomes familiar with the inner workings of Wechat. Wechat is fundamentally a social media… in store through Wechat pay are all possible. Many local and foreign companies already successful in China have official accounts on Wechat. An official … Wechat wallet. Many companies now also offer customer service through Wechat. Again, this is highly advisable as this is a service many Chinese consumers will now look for as it is quick and direct. Weibo is another popular social media app used across China. Think of Weibo as 'the Twitter of China'. Weibo is an open network site so users can see posts from anyone without being their friend or following them. Similar to Twitter, Weibo can be an excellent way to market your company by sharing the latest updates, offers, promotions etc. Your followers can also start to share your content helping your company's reputation spread by word of mouth.

[Ground Truth] ['Wechat', 'Wechat', 'Wechat', 'Wechat', 'Wechat', 'Wechat', 'Weibo', 'network site', 'Weibo', 'Weibo', 'app', 'Weibo', 'Baidu', 'Baidu', 'BAIDU']

GPT-4: ['Wechat', 'Weibo']

GPT-3.5: ['Taobao', 'Tmall', 'Wechat', 'Weibo']

Vicuna-13B-v1.5: ['WeChat (official accounts, WeChat pay, customer service)', 'Taobao (dominates shopping world)', 'Tmall (dominates shopping world)', 'Baidu (most popular search engine, brings WeChat)','Weibo (open network, excellent way to market, Twitter of China)']

SequenceTagger: ['taobao, 'tmall']

Ours (w/o INT): ['service', 'Wechat', 'Wechat', 'Wechat', 'Wechat', 'Wechat', 'Wechat', 'way', 'Weibo', 'network site', 'Weibo', 'Weibo', 'app', 'Weibo', 'Taobao', 'Tmall', 'Twitter', 'Twitter', 'China', 'China', 'China', 'China', 'China', 'China', 'China', 'Baidu', 'Baidu', 'BAIDU', 'Chinese', 'Chinese', 'Chinese']

Ours (w/o EXT): ['Weibo', 'Weibo', 'Wechat', 'Wechat', 'Weibo', 'BAIDU', 'Weibo', 'Baidu', 'Wechat', 'Twitter', 'Taobao', 'Tmall', 'Wechat', 'Wechat', 'Baidu', 'China', 'Wechat', 'China', 'app', 'Twitter', 'China', 'China', 'Chinese', 'network site', 'China', 'Chinese', 'China', 'Chinese', 'China', 'way', 'service']

Ours: ['Wechat', 'Wechat', 'Weibo', 'Weibo', 'Wechat', 'Wechat', 'Weibo', 'Wechat', 'Weibo', 'Wechat', 'Taobao', 'app', 'network site', 'Tmall', 'Twitter', 'BAIDU', 'Baidu', 'service', 'China', 'China', 'Twitter', 'Baidu', 'China', 'way', 'China', 'China', 'China', 'China', 'Chinese', 'Chinese', 'Chinese']

Table 12: Prediction result per different approaches. Note that our model uses thresholding for find proper points per query. In this result we show all ranking results.

	Matching Type	Search Mulitple Targets	Search Intention	External Knowledge-Augmented
Ctrl+F	Lexical	NO	Skimming	Manual
Regular Expression	Lexical	YES	Skimming	Manual
MRC	Semantic	YES	After Understanding	NO
KTRL+F	Semantic	YES	Skimming	Automatic

Table 13: Comparing characteristics of KTRL+F with other systems.

	Time(s)	# of Queries	# of visited Websites	Performance(List EM F1)
Ctrl+F	235(248)	7.47(8)	3.95(4.12)	58.64(61.79)
Regular Expression	265(275)	3.4(2)	3.54(4)	54.31(55.74)
KTRL+F plugin	211(217)	1.41(1.25)	1.08(1)	72.70(71.60)

Table 14: Evaluation table for comparing KTRL+F plugin with other systems. Averaged value is reported and median value are noted within bracket.

Search Intention	Query per System	Result
List the cities from California	**Ktrl+F :** List the cities from California	SAN JOSE , Calif . - Paramount to the ... they played smarter than they did Sunday in Anaheim , ... The Rangers signed 23-year-old defenseman Vince Pedrie out of Penn State, for whom he had 30 points in 39 games this season.
	Ctrl+F : [San jose, California, Anaheim]	SAN JOSE , Calif . - Paramount to the ... they played smarter than they did Sunday in Anaheim , ... The Rangers signed 23-year-old defenseman Vince Pedrie out of Penn State, for whom he had 30 points in 39 games this season.
	Regex: (SAN JOSE | California | Anaheim)	SAN JOSE , Calif . - Paramount to the ... they played smarter than they did Sunday in Anaheim , ... The Rangers signed 23-year-old defenseman Vince Pedrie out of Penn State, for whom he had 30 points in 39 games this season.
List all football teams	**Ktrl+F :** List all football teams	LIVERPOOL star Fabinho has been caught on camera appearing to sneeze on Chelsea 's Eden Hazard. Liverpool took back top spot in the Premier League after beating Chelsea at Anfield earlier today. The Reds now have four games ... leading Manchester City by ... "He's a fantastic player. Chelsea is ...
	Ctrl+F : [Liverpool, Chelsea, Manchester City]	LIVERPOOL star Fabinho has been caught on camera appearing to sneeze on Chelsea 's Eden Hazard. Liverpool took back top spot in the Premier League after beating Chelsea at Anfield earlier today. The Reds now have four games ... leading Manchester City by ... "He's a fantastic player. Chelsea is ...
	Regex: (LIVERPOOL | Chelsea | Manchester City)	LIVERPOOL star Fabinho has been caught on camera appearing to sneeze on Chelsea 's Eden Hazard. Liverpool took back top spot in the Premier League after beating Chelsea at Anfield earlier today. The Reds now have four games ... leading Manchester City by ... "He's a fantastic player. Chelsea is ...

Table 15: The figure above illustrates how each system handles the same search intention. It is worth noting that Ctrl+F and Regex require additional search engines to convert natural language search intentions, such as *"List the cities from California,"* into candidate keywords like *"Los Angeles, San Diego, San Jose, San Francisco, etc."* which consist of over a thousand cities. Moreover, there is no guarantee that these cities will appear on the web page. The highlighted text in yellow represents potential correct targets based on the query, while the red indicates possible false negative failures when using lexical search systems like Ctrl+F and Regex, which need to be highlighted.

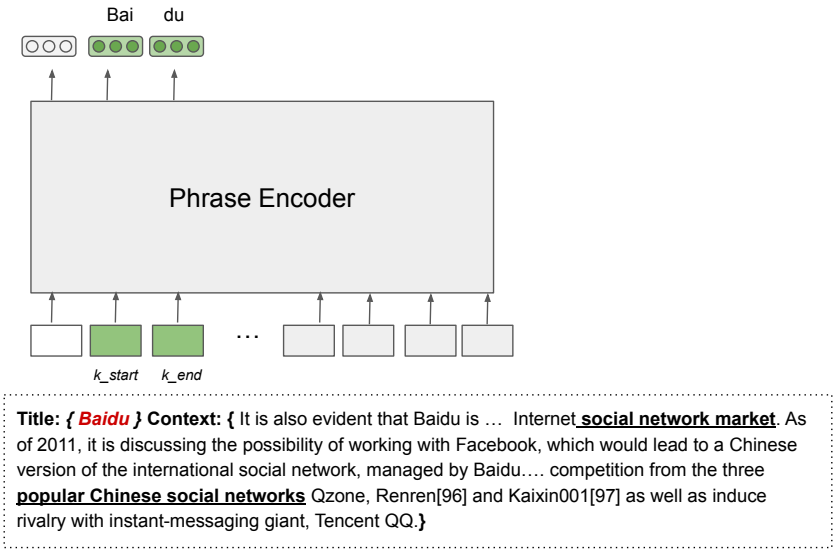


Figure 8: The figure demonstrates how to extract knowledge embedding, which is used for external knowledge for Knowledge-Augmented Phrase Retrieval. We utilize the frozen pre-trained phrase retrieval model (Lee et al., 2021), which shows good at encoding contextual information. The idea of using concatenated text with title and context and only extracting title embedding are following (Lee et al., 2023)

[Query] ***Companies founded by Bill Gates***
[Input Document]
That's a line remote co-workers often ask each other when they need to really discuss something, face-to-face or at least orally. But later this year, both of those software programs could find themselves sidelined by Slack. The makers behind the chat app announced yesterday that Slack users who are messaging each other will soon be able to have a voice call as well, and eventually a video call. No timeline has been given for either feature. Slack's rapid rise has already made it a darling of Silicon Valley. Just a year after its launch, investors valued the business chat app at over $1 billion. It was pegged at $2.8 billion as of last April, despite annual recurring revenues of just $25 million. That valuation is thanks to Slack's fast-growing, devoted customer base, which has skyrocketed from 500,000 daily active users in January 2015 to 2.3 million daily active users today. What percentage of these users are also Skype or Google Hangout users is impossible to say, but judging by purely anecdotal evidence from people in the tech and new media world, there's huge crossover. Most people who use Slack for business use a combination of Skype and Google Hangouts when they need to talk to someone face-to-face. But a majority of their time remains inside Slack, where they can write text messages to individual colleagues as well as set up team channels, upload gifs, and use special tools. Google is likely indifferent to Slack's rise. It makes almost all of its money from advertisements, and options like Google Hangouts are just there to keep users close to the search bar. But Skype is a different story. Skype, which is part of Microsoft, is mostly a two-trick app, and used for voice and video calls. Skype probably has more users that Slack right now, but some of its most valuable, paying "business" users are likely to fall into that "also using Slack" category. The company claimed 300 million monthly active users as of date, and had 4.9 million active daily users at the end of 2014, the latest daily data available. Skype for Business, the app's business-centric upgrade, is only a hair more sophisticated than regular Skype, which at the ripe old age of 12 both looks and feels dated. It has none of the bells and whistles that make Slack addictive in the first place. Adding voice and video calling to Slack could help businesses use one tool for nearly all communication purposes. Skype is owned by Microsoft, and accounts for a tiny percentage of the $93 billion in sales Microsoft brought in last year (they aren't broken out, but in 2013 Microsoft told Bloomberg Skype was set to bring in $2 billion in annual sales). Microsoft has been focusing on Windows Azure, an Amazon Web Services competitor, and revamping Skype is probably not a major priority. Microsoft did not respond to Quartz's requests for comment on this piece. But how many young companies are using Outlook? If internet history tells us anything, Skype is on its way to becoming the AOL of voice and video calling apps.
[Answer Entity] Microsoft
[LLAMA-2-Chat-7B]
Here are the spans from the article that correspond to the query "Companies founded by Bill Gates": * Microsoft * Skype
[LLAMA-2-Chat-13B]
Here are the spans that correspond to the query "Companies founded by Bill Gates": * Microsoft, which is owned by Bill Gates.

Table 16: Example of hallucination output of LLAMA-2.