Keyword Augmented Retrieval: Novel framework for Information Retrieval integrated with speech interface.

Anupam Puwar[a] and Rahul Sundar[b]

[a] - https://anupam-purwar.github.io/page/, Delhi, India

[b] - https://github.com/RahulSundar, Chennai, India

Abstract

Retrieving answers in a quick and low cost manner without hallucinations from a combination of structured and unstructured data using Language models is a major hurdle. This is what prevents employment of Language models in knowledge retrieval automation. This becomes accentuated when one wants to integrate a speech interface on top of a text based knowledge retrieval system. Besides, for commercial search and chat-bot applications, complete reliance on commercial large language models (LLMs) like GPT 3.5 etc. can be very costly. In the present study, the authors have addressed the aforementioned problem by first developing a keyword based search framework which augments discovery of the context from the document to be provided to the LLM. The keywords in turn are generated by a relatively smaller LLM and cached for comparison with keywords generated by the same smaller LLM against the query raised. This significantly reduces time and cost to find the context within documents. Once the context is set, a larger LLM uses that to provide answers based on a prompt tailored for Q&A. This research work demonstrates that use of keywords in context identification reduces the overall inference time and cost of information retrieval. Given this reduction in inference time and cost with the keyword augmented retrieval framework, a speech based interface for user input and response readout was integrated. This allowed a seamless interaction with the language model.

Information retrieval is one of the core subjects of the Information systems field and has been explored by various researchers with the twin objective of effective and efficient retrieval of information from massive collections of data. It has very important applications, including web based search engines, document retrieval systems and recommendation systems. The earliest works in this field around the Boolean model helped establish concepts for information retrieval queries, thus enabling precise searching. With the Vector Space Model being proposed by Salton *et al.* [1], this field gained a solid mathematical foundation as documents and queries were represented as high-dimensional vectors, enabling relevance ranking based on cosine similarity. Then came, Term Frequency-Inverse Document Frequency [2], shortly called TF-IDF which introduced the technique for term weighting, it proposed balancing importance of terms in a document relative to their frequency across the entire corpus. This was followed by the Probabilistic Information Retrieval Model [3] in the 1990s, which brought in the concept of uncertainty and document ranking based on likelihood estimates. On the web engine based information

retrieval two approaches namely BM25 (Best Matching 25) [4] and PageRank Algorithm [5], by Larry Page and Sergey Brin became popular with the latter leading to development of Google. Today, the Page rank algorithm is synonymous with web based search engines, it ranks web pages based on their link structure.
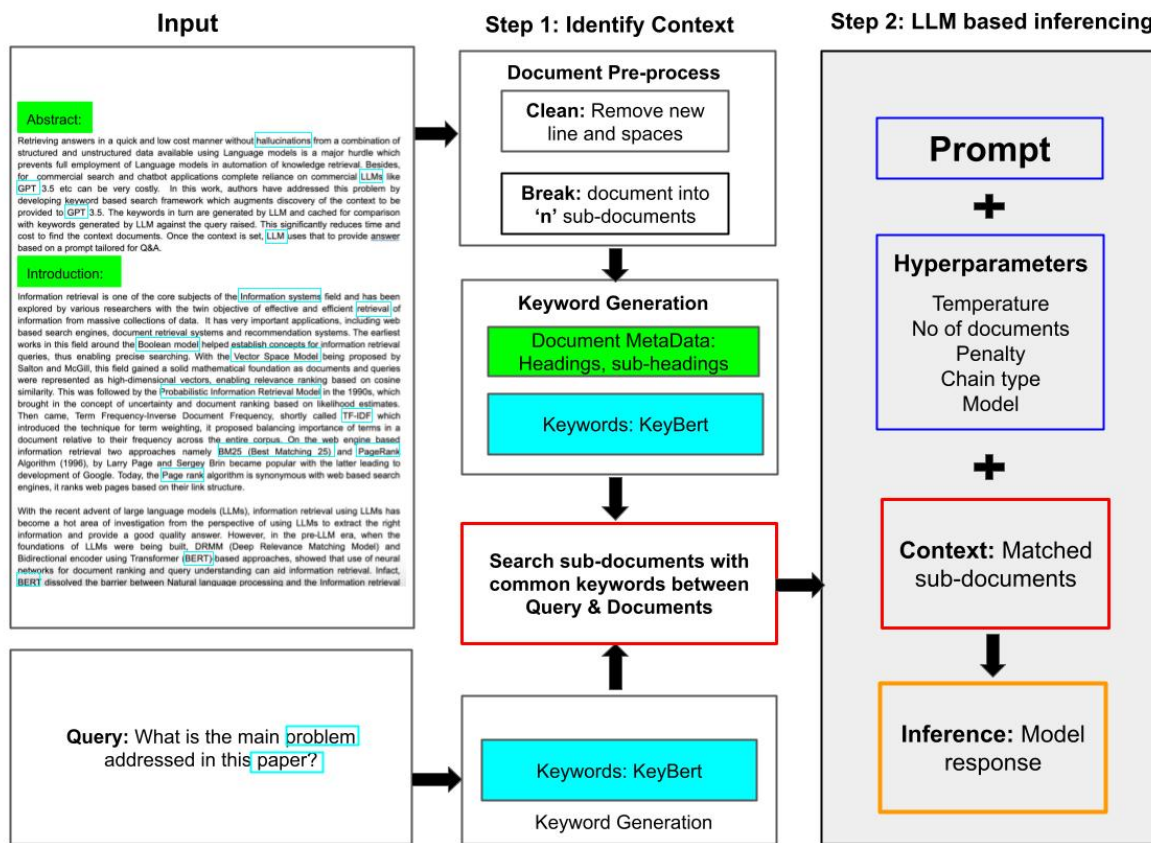

Figure 1: Workflow of KAR

With the recent advent of large language models (LLMs) [6], information retrieval using LLMs has become a hot area of investigation from the perspective of using LLMs to extract the right information and provide a good quality answer. However, in the pre-LLM era, when the foundations of LLMs were being built, DRMM (Deep Relevance Matching Model) [7] and Bidirectional encoder using Transformer (BERT) [8]-based approaches, showed that use of neural networks for document ranking and query understanding can aid information retrieval. In fact, BERT dissolved the barrier between Natural language processing and the Information retrieval (IR) field. BERT and other transformer based models like GPT [9], Google Flan-T5 [10] revolutionized the IR field by providing the power of real time interaction to users with their generative and context adapting capabilities.

Commercial LLMs like the ones from OpenAI, Anthropic have demonstrated promise in terms of providing good answers broadly on many topics. However, even the most sophisticated LLMs have hallucination issues and may end up providing unreasonable answers. Hallucination is the phenomenon where LLM generates false/incorrect or fabricated information. It has been identified as one the primary challenges while using LLM. It puts an unsuspecting user who is depending on these systems for information retrieval,

2

research and critical decision-making at huge risk. Hence, techniques like SelfCheckGPT are being proposed to evaluate the factuality of generated responses [11] using stochasticity in sampled responses. Besides, LLMs have limited context [12] and are not aware of latest developments due to the boundary created by the training data they are exposed to. Hence, they sometimes end up cooking up answers which are nothing but manifestations of words with highest probabilities being arranged in an easy to understand language [11]. Another challenge with the use of LLMs for IR tasks at scale is the high cost associated with commercial LLMs and high inference time/compute cost with open source LLMs. In this perspective, development of a technique which quickly provides a factually correct answer in an affordable manner becomes the need of the hour. In this work, authors propose a novel technique which augments the information retrieval using keywords and headings/labels available within the document.

1 Methodology

In the present work, we have devised a two step approach. In step 1, it focuses on understanding the question/user query and generating keywords from it followed by identification of right context from the document search space. Once the right context is identified, an answer is generated by the LLM using the context in step 2. The main contribution of the authors lies in the proposal to identify the right context using keywords generated by a transformer model KeyBert [13] (an open source Python library) and using these keywords alongwith in-text metadata like section headings to identify the right context from the document space in a quick and low cost manner.

1.1 Context identification

The document database consisting of all documents with text data as well as corresponding meta data is consumed by the model and this massive text data is broken down into smaller sub documents. The length of each sub-document is decided by the context length of the chosen LLM. All these sub-documents are stored and then keywords are extracted from each of them using keyBert library. For each sub-document, the corresponding heading as well as subheading data is also stored as keywords. Thus, it results in a dictionary consisting of each sub- document as index and corresponding keywords. Similar exercise is carried out for the query/question asked using keyBERT library to identify important keywords and query-keyword mapping is generated. Next, we compare the keywords corresponding to the query with keywords corresponding to the sub-documents and this helps identify relevant sub-documents for the question asked. This step replaces the usual step of finding out contextual information from document space using similarity between vector embeddings of query and the document.

1.2 LLM based inference

In the next step, contextual information pulled from the document database is consumed by the LLM along with a prompt to come up with a final answer. The length of answer is decided by the context length of the chosen LLM as well as the number of sub-documents which match the question asked. To minimize the randomness in generation of answer, LLM hyper-parameters viz. temperature is kept as 1e-6 and the LLM used here is text-
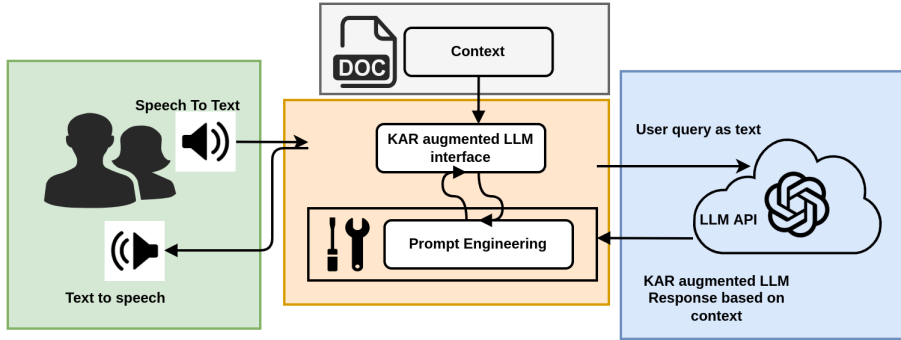
Figure 2: Speech interface integration with the KAR workflow.

davinci-003. Following prompt is used to bring out a context informed answer from the LLM.

"Answer the question based on the context below,and if the question can't be answered based on the context, say "I don't know"
\n\n Context: context\n\n

Question: query
\n Answer:"

1.3 Speech models integration

While the KAR is the foundation of the conversational interface, it was investigated how much additional time is added on integration of (i) speech to text models for query input, and (ii) text to speech conversion of the retrieved answer into the system (see figure 2). This is to understand if real time conversational document retrieval is possible through freely available `SpeechRecognition` [14] and google text-to-speech (`gTTS`) libraries.

2 Results

The proposed framework, Keyword augmented generation (KAR) consisting of steps 1 and 2 was applied to a document (MS word file) created from a web page. The code has been implemented using Python (Langchain [15], keyBERT [13], openAI's API [16], SpeechRecognition [14] and gTTS libraries). Inference carried out using context identified using vector embedding similarity between query and document space has been called regular and the one using the proposed Keyword augmented generation framework has been called KAR. Following are the results obtained for the information retrieval task using both approaches, the two primary metrics are inference time and accuracy of answers, as demonstrated by Table 1 below, KAR outperforms on both the metrics.

Accuracy has been determined based on the completeness of answer from the available information in the source document as well as a binary check on correctness of the answer, if the answer is incorrect, accuracy is marked as 0, if the answer provided by the approach is factually correct and uses the contextual information in the source document but fails to provide an answer based on all the available context, then the accuracy can lie between (0, 100]. Here, the KAR approach reduces inference cost as it does not require one to create embeddings for source document and query. Also, one does not need to store

Table 1: Inference time and accuracy metrics for regular and KAR approaches along with inference time for the speech to text and text to speech models.

Query	Answer length (Regular)	Time: Regular (in s)	Answer length (KAR)	Time:KAR (in s)	Time saved	Accuracy (KAR)	Accuracy (Regular)	Time(in s) (STT - Query)	Time(in s) (TTS - Answer)
tell me important facts about expandrank	305	4.034	369	1.569	61.10%	100%	100%	1.291	1.157
tell me important facts about positionrank	494	3.657	592	1.682	54.00%	100%	100%	1.108	2.656
what is positionrank	545	4.019	331	1.696	57.80%	100%	100%	0.669	1.675
what is page rank	311	2.341	282	2.005	**14.35**%	**85**%	**0**	0.650	2.830
tell me important facts about pagerank	365	2.020	600	2.583	**27.83**%	**95**%	**0**	0.8759	2.104
tell me important facts about page rank	365	2.281	391	1.352	**40.76**%	**75**%	**0**	1.009	1.218

source documents as vector databases, however one needs to store keyword-sub-document mapping information in a data-frame to be used during context identification.

Notably, integration of speech-based query input (STT) and an answer readout stage (TTS) only add approximately 1-2 seconds in inference, respectively as seen in table 1. However, if the queries were to go beyond a couple of sentences, then the STT response time is expected to go up further. So is the case for answer readout using TTS. Overall, the inference time for the entire speech based-KAR workflow is still of the same order as just regular text-only RAG. Cutting down the retrieval time for RAG based systems is thus crucial towards enabling such speech integration for more real-time conversational document retrieval systems.

3 Demo

The KAR framework has been implemented in the chat-bot developed for AIML Systems 2023. The chat-bot is created using Python and corresponding libraries like Langchain. Its user interface is created and hosted using Streamlit, as shown in Figure 3 below.

4 Conclusions

This work proposes a novel framework for Information Retrieval from documents using keywords generated by KeyBERT which uses BERT-embeddings as well use of document meta information viz. paragraph headings and subheadings to search and identify the right context corresponding to a user query. Here, a speech based interface was integrated with the keyword augmented retrieval workflow to obtain a more seamless interaction with the user. A speech to text model was used to convert the user speech into a text based query input for the LLM. The KAR augmented LLM responss were then read out using a text to speech model. It was noticed that the speech models along with
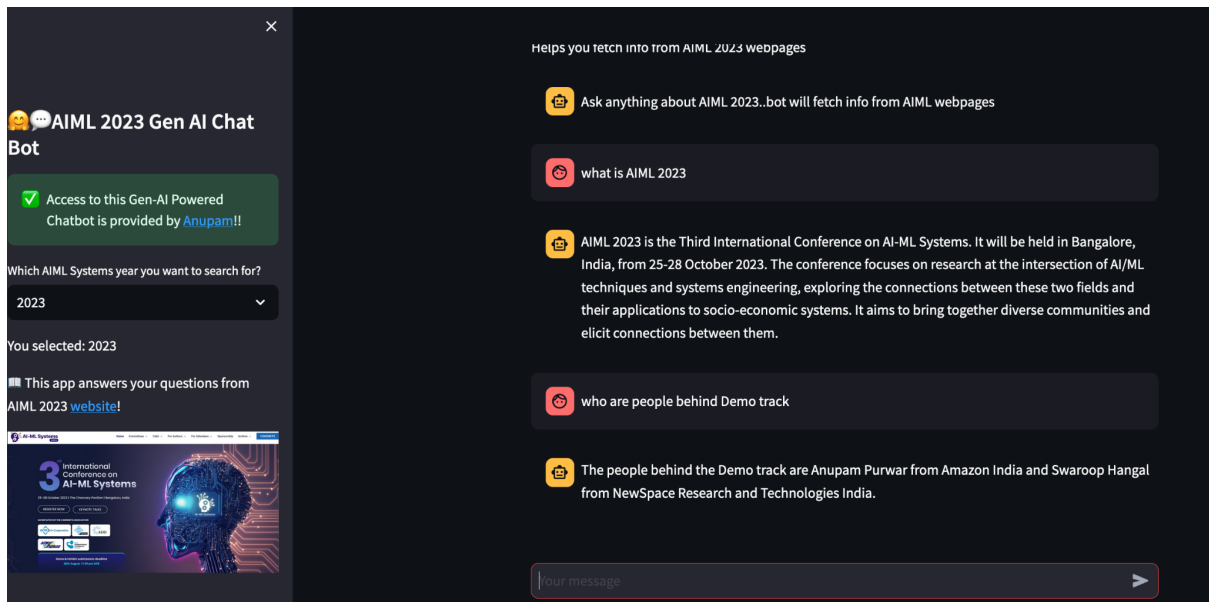
Figure 3: AIML Systems 2023: Generative AI powered bot

KAR together would still be comparable to the regular retrieval without speech. The results from implementation of the proposed KAR approach thus demonstrate better performance in terms of lower inference time, better accuracy and lower inferencing cost. In future, authors envisage to extend the framework to temporally resolve the data and make the time context also available to LLM for coming up with accurate answers with time sensitive data as well.

In addition, the speech interface could also be extended to allow voice cloning and a more human like response with more features including choice between a male and a female voice, and speech translation.

References

[1] A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[2] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[3] Norbert Fuhr. Probabilistic models in information retrieval. *The computer journal*, 35(3):243–255, 1992.

[4] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

[5] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[6] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang,

Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023.

[7] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, oct 2016.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pretraining of deep bidirectional transformers for language understanding, 2019.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[10] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

[11] Potsawee Manakul, Adian Liusie, and Mark JF Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*, 2023.

[12] Yucheng Li. Unlocking context constraints of llms: Enhancing context efficiency of llms with self-information-based content filtering, 2023.

[13] Maarten Grootendorst. Keybert: Minimal keyword extraction with bert., 2020.

[14] SpeechRecognition — pypi.org. https://pypi.org/project/SpeechRecognition/. [Accessed 26-09-2023].

[15] Oguzhan Topsakal and Tahir Cetin Akinci. Creating large language model applications utilizing langchain: A primer on developing llm apps fast. In *International Conference on Applied Engineering and Natural Sciences*, volume 1, pages 1050–1056, 2023.

[16] OpenAI Platform — platform.openai.com. https://platform.openai.com/docs/libraries. [Accessed 26-09-2023].