



EMERGING TRENDS

Emerging trends: a gentle introduction to RAG

Kenneth Ward Church , Jiameng Sun, Richard Yue , Peter Vickers , Walid Saba and Raman Chandrasekar 

Northeastern University, Boston, MA, USA

Corresponding author: Kenneth Ward Church; Email: k.church@northeastern.edu

(Received 12 June 2024; revised 12 June 2024; accepted 12 June 2024)

Abstract

Retrieval-augmented generation (RAG) adds a simple but powerful feature to chatbots, the ability to upload files just-in-time. Chatbots are trained on large quantities of public data. The ability to upload files just-in-time makes it possible to reduce hallucinations by filling in gaps in the knowledge base that go beyond the public training data such as private data and recent events. For example, in a customer service scenario, with RAG, we can upload your private bill and then the bot can discuss questions about your bill as opposed to generic FAQ questions about bills in general. This tutorial will show how to upload files and generate responses to prompts; see <https://github.com/kwchurch/RAG> for multiple solutions based on tools from OpenAI, LangChain, HuggingFace transformers and VecML.

Keywords: RAG; LLMs; chatbots; uploading files just-in-time; hallucinations

1. Introduction

This tutorial will start with simple chatbots, showing how to respond to prompts. These examples will introduce some opportunities for improvement such as hallucinations and timeliness. Retrieval-augmented generation (RAG) addresses some of these opportunities by adding the ability to upload files just-in-time.

2. Simple chat with openAI

This section will start with simple chats. Later, we will discuss RAGs that upload files just-in-time. Here is a simple example of a chat with OpenAI (and no uploading of files):^a

- **Input to ChatGPT:**
system: You are a helpful assistant.
user: Who won the world series in 2020?
assistant: The Los Angeles Dodgers won the World Series in 2020.
user: Where was it played?
- **Response from ChatGPT:**
The 2020 World Series was played at Globe Life Field in Arlington, Texas.

Details on how to run this example and other examples in this paper are posted on GitHub.^b Unfortunately, it may be difficult to replicate the exact outputs in this paper because chatbots are

^aThis example is based on: <https://platform.openai.com/docs/guides/text-generation>.

^b<https://github.com/kwchurch/RAG>



not always repeatable/deterministic. There are also concerns about ChatGPT's behavior changing over time (Chen, Zaharia, and Zou 2023).

The input to ChatGPT is a sequence of message objects. The message object contains two required fields: role and content. Examples of roles are *system*, *user*, and *assistant*.

2.1 Guard rails

The following example is like the example above, except that we ask ChatGPT to predict the future:

- **Input to ChatGPT:**
system: You are a helpful assistant.
user: Who won the world series in 2035?
- **Response from ChatGPT:**
 I'm sorry, but I am unable to provide real-time information or details about future events as I exist in the present. You may want to check the latest sports news or websites for updates on the World Series winners in 2035.

ChatGPT is constructed with guard rails designed to reduce "hallucinations." "Hallucination" is a technical term for "making stuff up." Chatbots have a tendency to spread misinformation. Guard rails help in easy cases like this, though it is not hard for an adversary to find small modifications to input prompts that will crash through the guard rails.

2.2 Timeliness

The next example replaces "2035" with "most recent." The point of this modification is to make it clear that we are using an old model that was trained in 2021:

- **Input to ChatGPT:**
system: You are a helpful assistant.
user: Who won the most recent world series?
- **Response from ChatGPT:**
 The Atlanta Braves won the most recent World Series in 2021.

We used an old model because OpenAI charges more for more recent models. But even so, since training large language models (LLMs) is expensive, it is prohibitively expensive for OpenAI to continuously update their models. The following chat shows that phrases like "most recent" call out differences between training time and inference time. According to Google and Wikipedia, the Texas Rangers won the most recent World Series (in 2023), but ChatGPT-3 says the Atlanta Braves won (in 2021). It is easier to keep Google and Wikipedia up to date than ChatGPT. In Subsection 3.2, we will show how RAG can work around the need to continuously update models.

2.3 Hallucinations when asked to discuss content that goes beyond training data

The next example was designed to elicit misinformation. The prompt presupposes that we have been talking about an (unspecified) paper on psycholinguistics. One might hope there would be a guard rail that would catch this prompt and respond, "I'm sorry, but I don't know what paper you are referring to." As we will see below, there is a guard rail that catches similar prompts. Unfortunately, in this case, the response below starts with a pivot to a different question: "What is Psycholinguistics?" After that pivot, the response continues by hallucinating what might be in the unspecified paper.

- **Input to ChatGPT:**
system: You are a helpful assistant.
user: Please summarize the paper on psycholinguistics.

- **Response from ChatGPT:**

Sure! Psycholinguistics is a field that studies the psychological processes involved in language comprehension and production. The paper discusses various topics in psycholinguistics, such as how language is processed in the brain, how we acquire language skills, and how language is used in communication. It also explores issues related to bilingualism, language development in children, and the impact of language disorders on individuals. Overall, the paper provides insights into how the human mind processes and understands language.

Robustness is a challenge for guard rails. A small change to the prompt can have a large impact. Consider the following chat where we replace a period (in the chat above) with a question mark (in the chat below). The guard rails are more effective with the question mark, at least in this particular instance:

- **Input to ChatGPT:**

system: You are a helpful assistant.

user: Please summarize the paper on psycholinguistics?

- **Response from ChatGPT:**

I would need more specific information to provide an accurate answer. There are numerous papers on the topic of psycholinguistics written by various authors. If you have a specific paper or author in mind, please provide more details so I can assist you better.

More generally, hallucinations are a challenge for LLMs when asked to discuss content that goes beyond training data. A lawyer once found himself in trouble when he trusted ChatGPT to help write a brief.^c

There was just one hitch: No one – not the airline’s lawyers, not even the judge himself – could find the decisions or the quotations cited and summarized in the brief.

A study on legal hallucinations from Yale and Stanford (Dahl *et al.* 2024) reports that the problem is “alarmingly prevalent, occurring between 69% of the time with ChatGPT 3.5 and 88% with Llama 2, when these models are asked specific, verifiable questions about random federal court cases.”

3. Opportunities for RAG

The examples above illustrate three opportunities for improvement:

More robust guard rails: Too easy to crash through existing guard rails (Nasr *et al.* 2023),

Pivots/Hallucinations: Chatbots tend to pivot and/or hallucinate when asked to discuss content that goes beyond the training set such as an unspecified paper (Subsection 2.3), and

Timeliness: Training time \neq inference time (Subsection 2.2)

Critics of large language models (LLMs) and chatbots bring up many classic topics in Philosophy of Language, Artificial Intelligence and Creative Writing such as:^d

1. Hallucinations and Misinformation: Fact-checking, Grounding, Attribution
2. Knowledge Acquisition: Timeliness, Plagiarism^e (Nasr *et al.* 2023)
3. Knowledge Representation: Lexical Semantics, Ontologies, World Knowledge, Semantics of Time and Space, Technical Terminology, Domain Specific Knowledge

^c<https://www.nytimes.com/2023/05/27/nyregion/avianca-airline-lawsuit-chatgpt.html>

^dSee slide 4 of https://github.com/Tongji-KGLLM/RAG-Survey/blob/main/assets/RAG_Slide_ENG.pdf for another list of opportunities for improvement.

^e<https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html>

4. Reference: Co-reference, Given/New, Use/Mention,^f Intensional Logic,^g Possible Worlds,^h Belief^f
5. Discourse Structure: Grice's Maxims (Grice 1975), Perspective, Character Development,^j Plot (and Plot Twists)
6. Problem Solving: Planning (Kautz and Allen 1986), Common Sense Reasoning (McCarthy 1969), Problem Decomposition
7. Explanation (Ribeiro, Singh, and Guestrin 2016)

RAG addresses some of these opportunities by adding a feature to upload documents just-in-time. Another motivation for uploading documents just-in-time involves private data. LLMs are trained on massive amounts of public data, but most documents are private. Suppose, for example, we are building a chatbot for customer support. Users will have questions about their bills. If we can upload their private bills just-in-time, then we can address their questions in the context of their bill. Otherwise, the chatbot can do little more than discuss generic FAQs about billing questions in general, but not specifically about their bill.

3.1 Recipe

Four implementations of RAG are posted on our GitHub: (1) [src/OpenAI/Rag.py](#), (2) [src/LangChain/Rag.py](#), (3) [src/transformers/Rag.py](#), and (4) [src/VecML/Rag.py](#). They all use the following recipe:

1. Upload one or more files
2. Parse files into chunks
3. Embed chunks as vectors in a vector space
4. Index vectors with an approximate nearest neighbor (ANN) algorithm such as ANNOY^k or FAISS^l (Johnson, Douze, and Jégou 2019; Douze *et al.* 2024)
5. Retrieve chunks near prompt (using ANN)
6. Generate responses

All four RAG.py programs share the same interface. They take a list of files to upload on the command line. Input prompts are read from standard input, and output responses are written to standard output. The programs are short, less than 100 lines; the VecML program is particularly short:

```

1 from vecml import vecml
2 import sys,os
3
4 chatName = sys.argv[1]
5
6 vecml.init(os.environ[' VECML_API_KEY '], "us-west ")
7 vecml.create_chat(chatName, sys.argv[1:])
8
9 for prompt in sys.stdin:
10     print (vecml.chat(chatName, prompt))

```

^f<https://plato.stanford.edu/entries/quotation/>

^g<https://plato.stanford.edu/entries/logic-intensional/>

^h<https://plato.stanford.edu/entries/possible-worlds/>

ⁱ<https://plato.stanford.edu/entries/belief/>

^j<https://industrialscripts.com/archetypes-of-characters/>

^k<https://github.com/spotify/annoy>

^l<https://github.com/facebookresearch/faiss>

Some of the shorter RAG.py programs hide various steps in the recipe above behind APIs running on the cloud.

There are also a few programs, chat.py. These programs read prompts from standard input and write responses to standard output, but unlike RAG.py programs, the chat.py programs do not upload files.

3.2 Timeliness and a simple example of RAG

This paper will not attempt to discuss all of the opportunities raised above, except to suggest that RAG can help with a few of them. Consider the timeliness opportunity in Subsection 2.2; obviously, users do not want “day-old donuts” like who won the World Series when the bot was trained a few years ago.

On the web, information ages at different rates. Before the web, news was typically relevant for a news cycle, though breaking news was already an exception. These days, news typically ages more quickly than it used to, but even so, the half-life of news is probably a few hours. Some information sources age more quickly such as stock prices, and other information sources age more slowly such as the academic literature. Within the academic literature, some venues (arXiv and conferences) age more quickly than others (archival journals). Crawlers for search engines like Google and Bing need to prioritize crawling by trading-off these aging properties with demand. Since crawlers cannot afford to crawl all pages all the time, they need to crawl pages often enough to keep up with aging, but if that is not possible, then they should prioritize pages that are more likely to be requested.

How does RAG help with timeliness? Rather than use LLMs as is, and attempt “closed-book question-answering” with an “out-of-date book,” RAG uses R (retrieval/search) and A (augmentation) to update “the book” (knowledge base) just-in-time. For example, if we want to know who won the most recent World Series, then it might help to do a search for documents on that topic such as this.^m RAG makes it possible to upload files before generating responses to the prompt. On the GitHub,ⁿ we work through this example in detail:

```
1 echo 'Who won the world series in 2023?' |
2 src/OpenAI/RAG.py sample_files/World_Series/*.pdf
```

Without RAG, an LLM trained on 2021 data would likely hallucinate when asked about 2023. RAG fills in gaps in the knowledge base by uploading a pdf file, sample_files/World_Series/*.pdf, a version of the text from footnote^m, and uses that content to update the knowledge base on this topic from 2021 to cover 2023. In this way, RAG works around the need to continuously retrain the bot, by creating a just-in-time mechanism for updating the knowledge base.

3.3 References to background context and RAG

RAG can also help with references to the discourse context. If we upload a paper on psycholinguistics, then it can address the prompt: “Please summarize the paper on psycholinguistics.”

```
1 echo 'Please summarize the paper on psycholinguistics.' |
2 src/OpenAI/RAG.py sample_files/papers/*.pdf
```

The directory, sample_files/papers, contains two pdf files for Church and Hanks (1990) and Lin (1998). We would like to ask the bot to summarize (Church and Hanks 1990), but since the bot is unlikely to understand citations, we will refer to the two papers with phrases such as

^mhttps://en.wikipedia.org/wiki/List_of_World_Series_champions

ⁿ<https://github.com/kwchurch/RAG/tree/main?tab=readme-ov-file#rag>

Table 1. RAG summaries are longer than tl;dr summaries from Semantic Scholar

Id: Title	OCR	tl;dr Summary	RAG summary
ACL:J90-1003: Word Association Norms, Mutual Information, and Lexicography (Church and Hanks 1990)	✓	The proposed measure, the association ratio, estimates word association norms directly from computer readable corpora, making it possible to estimate norms for tens of thousands of words	The paper discusses the importance of word association norms in psycholinguistic research, particularly in lexical retrieval. It introduces the concept of the association ratio as a measure to estimate word association norms directly from computer-readable corpora. The paper extends the psycholinguistic notion of word association norms toward an information theoretic definition of mutual information, providing statistical calculation that can be applied effectively. The standard method of obtaining word association norms through testing subjects is deemed costly and unreliable compared to the proposed measure utilizing computer data
ACL:C98-2122: Automatic Retrieval and Clustering of Similar Words (Lin 1998)	✓	A word similarity measure based on the distributional pattern of words allows a thesaurus to be constructed using a parsed corpus and shows that the thesaurus is significantly closer to WordNet than Roget Thesaurus is	The paper discusses clustering similar words and evaluates clusters based on their ability to recover data items that are removed from the input corpus. It also indirectly evaluates collocations and their associated scores by their use in parse tree distributional clustering of English words. The paper was presented at ACL93 and authored by Gerda Ruge in 1992. The future work in clustering similar words is briefly mentioned in Section 4, while related work and the contributions of the paper are summarized in Section 5
ArXiv: 2005.11401: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (Lewis <i>et al.</i> 2020)	✗	A general-purpose fine-tuning recipe for retrieval-augmented generation (RAG) – models that combine pretrained parametric and nonparametric memory for language generation and finds that RAG models generate more specific, diverse, and factual language than a state-of-the-art parametric-only seq2seq baseline	The paper on RAG discusses the use of retrieval-augmented generation (RAG) for question answering (QA). RAG directly minimizes the negative log-likelihood of answers and is compared to traditional extractive QA methods and Closed-Book QA approaches. RAG’s retriever is initialized using DPR’s retriever, and it achieves results comparable to the DPR QA system. RAG is able to generate answers without the need for specialized pre-training like “salient span masking.” Additionally, RAG demonstrates high accuracy in classifying claims as true or false based on evidence it retrieves, achieving results within 2.7% of a model that uses gold evidence sentences. In analyzing RAG’s performance, the overlap in article titles between the documents retrieved by RAG and the gold evidence in FEVER dataset is calculated

- the paper on psycholinguistics
- the paper on clustering

The program listing above produces the RAG summaries in Table 1. Summaries from RAG are impressive, at least on first impression. However, on further reflection, after reading a number of such summaries, it becomes clear that there are many opportunities for improvement.

RAG summaries are probably worse than summaries from previous technologies such as tl;dr (too long; did not read) summaries from Semantic Scholar (S2).^o Table 1 compares RAG summaries with tl;dr summaries; S2 tl;dr summaries are shorter, and probably more to the point, though it is hard to define “more to the point” precisely.

^o<https://www.semanticscholar.org/paper/9e2caa39ac534744a180972a30a320ad0ae41ea3>

score	citationCount	Paper	Authors	year	More like this	Compare & Contrast	ProNE-s	SciNCL	Specter	GNN
865		Personalizing Search via Automated Analysis of Interests and Activities	J. Teevan, S. Dumais, E. Horvitz	2005	Similar to this	Compare & Contrast	1.0	1.0	1.0	1.0
0.997	564	Implicit user modeling for personalized search	Xuehua Shen, Bin Tan, ChengXiang Zhai	2005	Similar to this	Compare & Contrast	0.997	0.938	0.95	0.979
0.997	84	Beyond the Commons: Investigating the Value of Personalizing Web Search	J. Teevan, S. Dumais, E. Horvitz	2005	Similar to this	Compare & Contrast	0.997	0.919	0.933	0.99
0.996	203	Interest-based personalized search	Zhongmin Ma, Gautam Pant, O. Sheng	2007	Similar to this	Compare & Contrast	0.996	0.923	0.942	0.937

Figure 1. The query (top line) is followed by recommendations.

3.4 Scenario: compare and contrast recommendations

Figure 1 shows a recommendation engine.^p We would like to provide users with explanations that work for them. Users probably do not want to know how the different recommendation engines work. It is probably not helpful to discuss details such as deep networks, embeddings, approximate nearest neighbors, and spectral clustering on citation graphs. Rather, we propose to offer users buttons like the “compare & contrast” buttons in Fig. 1. Users can click on one of these buttons if they want to know why a paper is being recommended. The implementation of these buttons is based on RAG. In this way, RAG can be used to address some of the explanation opportunities mentioned above. If a user clicks on one of these buttons, the relevant papers will be uploaded to a RAG system and then the RAG system will be prompted to not only summarize the relevant papers one at a time but also to compare and contrast pairs of papers at a time.^q

3.5 Compare and contrast with VecML.com

Our GitHub provides several versions of RAG.py. In addition to the OpenAI version mentioned above, there is also a version based on [VecML.com](#).^r Here is an example that uses the [VecML.com](#) version of RAG.py:

```
1 echo 'Please summarize the paper on psycholinguistics.' >/tmp/x
2 echo 'Please summarize the paper on clustering.' >>/tmp/x
3 echo 'What are the similarities between the two papers?' >>/tmp
  /x
4 echo 'What are the differences?' >>/tmp/x
5 src/VecML/RAG.py sample_files/papers/*.pdf </tmp/x
```

The VecML code for RAG.py^s is even shorter than the OpenAI code for RAG.py.^t As mentioned above, all of the RAG.py programs take one or more papers on the command line and upload them to the cloud. Then they read prompts from standard input and print responses to standard output.

If we run the VecML RAG.py program on two ACL papers, Church and Hanks (1990), Lin (1998), using the prompts below:

1. Please summarize the paper on psycholinguistics.
2. Please summarize the paper on clustering.
3. What are the similarities between the two papers?
4. What are the differences?

^p<https://www.semanticscholar.org/api-gallery/better-together>

^q An implementation of the compare & contrast button is posted here: https://github.com/kwchurch/Better_Together_Resources/blob/main/cgi-bin/api/compare_and_contrast.

^r<https://github.com/kwchurch/RAG?tab=readme-ov-file=vecml>

^s<https://github.com/kwchurch/RAG/blob/main/src/VecML/RAG.py>

^t<https://github.com/kwchurch/RAG/blob/main/src/OpenAI/RAG.py>

We obtain the following results:

Summarize (Church and Hanks 1990): *The paper on psycholinguistics discusses the importance of word association norms in psycholinguistic research, particularly in the area of lexical retrieval. It mentions that subjects respond quicker to words that are highly associated with each other. While noun–noun word associations like “doctor/nurse” are extensively studied, less attention is given to associations among verbs, function words, adjectives, and other non-nouns. The paper concludes by linking the psycholinguistic notion of word association norms to the information-theoretic concept of mutual information, providing a more precise understanding of word associations.*

Summarize (Lin 1998): *The paper discusses a triangulation approach for clustering concordance lines into word senses based on usage rather than intuitive meanings. It highlights the superfluity of defining a word measure for clustering words without explicit preprocessing tools such as Church’s parts program or Hindle’s parser. The paper briefly mentions future work on clustering similar words and reviews related work while summarizing its contributions.*

Similarities: *The similarities between the two papers include a focus on analyzing language data, using distributional patterns of words, evaluating similarity measures for creating a thesaurus, and discussing the importance of smoothing methods in language processing tasks.*

Differences: *The differences between the two thesaurus entries can be measured based on the cosine coefficient of their feature vectors. In this case, the differences are represented in the relationships between the words listed in each entry. For example, in the given entries, “brief (noun)” is associated with words like “differ,” “scream,” “compete,” and “add,” while “inform” and “notify” are related to each other in the second entry. These associations indicate the semantic relationships and differences between the words in each entry.*

Like summaries, output from the compare-and-contrast button looks impressive on first impression, but on further reflection, the responses can be long-winded, and sometimes seem to miss the point. It is not clear that the bot “understands” concepts such as similarities and differences. The bot’s discussion of differences is not addressing differences between the two papers, but rather differences within one of the papers among lexical items. We have obtained slightly better responses by rephrasing the prompts in various ways, but we will not discuss such complications here since this tutorial is intended to provide a gentle introduction (as opposed to SOTA-chasing).

That said, as an area chair and senior area chair for many conferences, we have seen many reviews from unmotivated (and unqualified) reviewers. It is not clear that RAG is worse than bad reviews, though there are obvious ethics concerns with using RAG (and unmotivated/unqualified reviewers) for high-stakes decisions with long-term implications for careers.

4. RAG is not magic

4.1 Problem decomposition

Uploading documents just-in-time addresses some of the opportunities raised above, but not all. Consider problem decomposition. Chatbots can easily add two small numbers, but they cannot add two large numbers. Children have the reverse problem. They struggle with memorizing multiplication and addition tables, but they have little trouble decomposing the sum of two big numbers into more manageable tasks. You do not have to teach a child fancy principles like superposition. They just get it.

A popular workaround to this problem is Chain-of-Thought (CoT) Prompting (Wei *et al.* 2022). Since chatbots struggle with decomposing larger tasks into more manageable subtasks, the community has discovered that chatbots will be more successful if prompts come already predecomposed. In this way, CoT is seen by most researchers as an advance, but it might be more appropriate to think of it as a workaround. Just as a parent might cut up a child’s dinner

to prevent choking incidents, so too, CoT Prompting cuts up prompts into more manageable bite-size chunks.

In short, chatbots are not magic. Chatbots will be more successful if we spoon-feed them. RAG spoon-feeds them by inserting the relevant content into the input before invoking the response, and CoT Prompting spoon feeds them by cutting up prompts into bite-size pieces to prevent hallucinations.

4.2 Opportunities for improvement

There are a couple of opportunities to improve the example above:

1. OCR errors: garbage in → garbage out
2. KISS (keep it simple, stupid):
 - (a) It is safer to process fewer files at a time, and
 - (b) To decompose prompts into smaller subtasks (CoT reasoning)

Consider OCR errors. RAG tends to process documents in very simple ways, typically as a sequence of chunks, where each chunk is a sequence of no more than 512 subword units. In fact, documents are much more complicated than that. Many documents contain tables, figures, equations, references, footnotes, headers, footers and much more. There are many OCR errors in older pdf files on the ACL Anthology, as illustrated in Table 2.

4.3 OCR errors and spaCy summarizations

It is remarkable how well RAG does on documents with OCR errors. Older technologies such as spaCy^u are less robust to OCR errors, as discussed on our GitHub.^v Table 2 shows that OCR errors are more challenging for spaCy than RAG.

5. Where is the RAG literature going?

The literature on RAG has been exploding recently. There are 381 references in Zhao *et al.* (2024), of which 168 (44%) were published in 2023 or 2024. Given the volume, veracity (and recency) of the literature on RAG, it is difficult to see where it is going at this early point in time, but the list below suggests that benchmarking is a hot topic. The field is convinced that RAG is an important advance, but the field is still trying to figure out how to measure progress.

Surveys: Gao *et al.* (2023), Zhao *et al.* (2024)^w

Tutorials: DSPy^x and some even more gentle introductions to RAG than this paper: *LangChain Explained in 13 Minutes*,^y *Langchain RAG Tutorial*,^z *Learn RAG From Scratch*^{aa} and *Simple Local Rag*^{ab}

^u<https://spacy.io/>

^v<https://github.com/kwchurch/RAG?tab=readme-ov-file#summarizing-with-spacy>

^w<https://github.com/PKU-DAIR/RAG-Survey>

^x<https://dspy-docs.vercel.app/>

^yShort Video (13 minutes): <https://www.youtube.com/watch?v=aywZrzNaKjs> GitHub: <https://github.com/rabbitmetrics/langchain-13-min>

^zVideo (2 hours): <https://youtu.be/tcqEUSNCn8I> GitHub: <https://github.com/pixegami/langchain-rag-tutorial>

^{aa}Video: <https://www.youtube.com/watch?v=sVcwVQRHic8> GitHub: <https://github.com/langchain-ai/rag-from-scratch>

^{ab}Video (5 hours): https://www.youtube.com/watch?v=qN_2fnOPY-M; GitHub: <https://github.com/mrdbourke/simple-local-rag>

Table 2. OCR errors are more challenging for spaCy than RAG

Id: Title	OCR	Spacy summary
ACL:J90-1003: Word Association Norms, Mutual Information, and Lexicography (Church and Hanks 1990)	✓	The proposed measure, the association ratio, estimates word association norms directly from computer readable corpora, waki, g it possible to estimate norms for tens of thousands of words. (Meyer, Schvaneveldt, and Ruddy, 1975, p. 98) Much of this psycholinguistic research is based on empirical estimates of word association norms such as Palermo and Jenkins (1964), perhaps the most influential study of its kind, though extremely small and somewhat dated
ACL:C98-2122: Automatic Retrieval and Clustering of Similar Words (Lin 1998)	✓	Unlike sim, simninal, and simHiner, they only 770 210g P(c), $\sim \text{simwN}(w_1, w_2) = \max_c \sim eS(w \sim) \text{Ac}2eS(w_2)$ ($\max_{c \sim} \text{super}(c \sim) \text{nsuper}(c_2)$ $\log P(c_1) + \log P(c_2) ! 21R(\sim) \text{nR}(w_2) \text{simRoget}(W_1, W_2) = R(w_x) + R(w_2) $ where S(w) is the set of senses of w in the WordNet, super(c) is the set of (possibly indirect) superclasses of concept c in the WordNet, R(w) is the set of words that belong to a same Roget category as w. Figure 2: Word similarity measures based on WordNet and Roget make use of the unique dependency triples and ig~ Contextual word similarity and estimation from sparse data
ArXiv: 2005.11401: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (Lewis <i>et al.</i> 2020)	✗	We introduce RAG models where the parametric memory is a pretrained seq2seq model and the nonparametric memory is a dense vector index of Wikipedia, accessed with a pretrained neural retriever. For language generation tasks, we find that RAG models generate more specific, diverse, and factual language than a state-of-the-art parametric-only seq2seq baseline

Benchmarking: CRAG,^{ac} RGB^{ad} (Chen *et al.* 2024), KILT^{ae,af} (Petroni *et al.* 2021), ARES^{ag} (Saad-Falcon *et al.* 2023), TruLens,^{ah} CRUD-RAG^{ai} (Lyu *et al.* 2024), MIRAGE^{aj} (Xiong *et al.* 2024), EXAM (Sander and Dietz, 2021), CLAPNQ^{ak} (Rosenthal *et al.* 2024), Retrieval-QA,^{al} Hallucination Leaderboard,^{am} RAGAs^{an} (Es *et al.* 2024), HaluEval (Li *et al.* 2023), PHD^{ao} (Yang, Sun, and Wan 2023), and RAGTruth^{ap} (Wu *et al.* 2023)

Benchmarks focus the research community on specific opportunities. For example, the RGB Benchmark is designed to address four opportunities in English (and Chinese) datasets:

1. Noise robustness (includes timeliness opportunities),
2. Negative rejection (guard rails),
3. Information integration (answering questions by combining two or more documents), and

^{ac}<https://www.aicrowd.com/challenges/meta-comprehensive-rag-benchmark-kdd-cup-2024>

^{ad}<https://github.com/chen700564/RGB>

^{ae}<https://github.com/facebookresearch/KILT>

^{af}<https://huggingface.co/datasets?search=kilt>

^{ag}<https://github.com/stanford-futuredata/ARES>

^{ah}<https://github.com/truera/trulens>

^{ai}https://github.com/IAAR-Shanghai/CRUD_RAG

^{aj}<https://github.com/Teddy-XiongGZ/MIRAGE>

^{ak}<https://github.com/primeqa/clapnq>

^{al}<https://github.com/myscale/Retrieval-QA-Benchmark>

^{am}<https://huggingface.co/spaces/vectara/Hallucination-evaluation-leaderboard>

^{an}<https://github.com/explodinggradients/ragas>

^{ao}<https://github.com/maybenotime/PHD>

^{ap}<https://github.com/ParticleMedia/RAGTruth>

4. Counterfactual robustness (robustness to factual errors in documents)

CLAPNQ (Cohesive Long-form Answers from Passages in Natural Questions) addresses guard rails plus four new opportunities:

1. Faithfulness (answer must be grounded in gold passage),
2. Conciseness (answer must be short, excluding information unrelated to gold answer),
3. Completeness (answer must cover all information in gold passage)
4. Cohesiveness

CLAPNQ is based on the natural questions (NQ) benchmark (Kwiatkowski *et al.* 2019). Another recent paper on NQ (Cuconasu *et al.* 2024) shows that performance improves with some conditions and degrades with others. They considered (1) gold context (from benchmark), (2) relevant documents that contain correct answers, (3) related documents that do not contain correct answers, and (4) irrelevant random documents. It may not be surprising that (3) degrades performance, but their main (surprising) result is (4) improves performance.

The benchmarking approach is likely to make progress on many of these topics, though we have concerns about guard rails (and hallucinations). Bots tend to go “off the rails” when they fail to find evidence. It is impressive how much progress the field has made with zero-shot reasoning recently, but even so, reasoning in the absence of evidence is challenging. Failure to find fallacies are hard.

Maybe we can make progress on hallucinations by measuring average scores on benchmarks, or maybe we need a different approach such as theoretical guarantees (from theoretical computer science) or confidence intervals (from statistics). Statistics distinguishes the first moment (expected value) from the second moment (variance). When there is little evidence to support a prediction, confidence intervals are used to prevent hallucinations. Perhaps benchmarks should become more like calibration in statistics. When we have such and such evidence, how likely is the bot to be correct? Guard rails should be deployed when confidence fails to reach significance.

6. Conclusions

This tutorial (and the accompanying GitHub) showed a number of implementations of chatbots and RAG using tools from OpenAI, LangChain, HuggingFace, and VecML. RAG improves over chatbots by adding the ability to upload files just-in-time. Chatbots are trained on massive amounts of public data. By adding the ability to upload files just-in-time, RAG addresses a number of gaps in the chatbot’s knowledge base such as timeliness, references to background knowledge, private data, etc. Gaps in the knowledge base can lead to hallucinations. By filling in many of these gaps just-in-time, RAG reduces the chance of hallucinations.

References

- Chen J., Lin H., Han X. and Sun L. (2024). Benchmarking large language models in retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence* 38(16), 17754–17762.
- Chen L., Zaharia M. and Zou J. (2023). How is ChatGPT’s behavior changing over time? arXiv preprint arXiv:2307.09009.
- Church K.W. and Hanks P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics* 16(1), 22–29.
- Cuconasu F., Trappolini G., Siciliano F., Filice S., Campagnano C., Maarek Y., Tonello N. and Silvestri F. (2024). The power of noise: redefining retrieval for rag systems. ArXiv preprint abs/2401.14887.
- Dahl M., Magesh V., Suzgun M. and Ho D.E. (2024). Large legal fictions: profiling legal hallucinations in large language models. *Journal of Legal Analysis* 16(1), 64–93. arXiv preprint arXiv: 401.01301.
- Douze M., Guzhva A., Deng C., Johnson J., Szilvasy G., Mazaré P.-E., Lomeli M., Hosseini L. and Jégou H. (2024). The FAISS library. arXiv:2401.08281.

- Es S., James J., Espinosa Anke L. and Schockaert S. (2024). *RAGAs: automated evaluation of retrieval augmented generation*. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, St. Julians, Malta. Association for Computational Linguistics, pp. 150–158.
- Gao Y., Xiong Y., Gao X., Jia K., Pan J., Bi Y., Dai Y., Sun J. and Wang H. (2023). Retrieval-augmented generation for large language models: a survey. arXiv preprint arXiv:2312.10997.
- Grice H.P. (1975). Logic and conversation. In Cole P. and Morgan J.L. (eds), *Syntax and Semantics*, Speech Acts, Vol. 3. New York: Speech Acts, Academic Press, pp. 41–58.
- Johnson J., Douze M. and Jégou H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7(3), 535–547.
- Kautz H.A. and Allen J.F. (1986). *Generalized plan recognition*. In *AAAI Conference on Artificial Intelligence*, Philadelphia, PA.
- Kwiatkowski T., Palomaki J., Redfield O., Collins M., Parikh A., Alberti C., Epstein D., Polosukhin I., Devlin J., Lee K., Toutanova K., Jones L., Kelcey M., Chang M.-W., Dai A.M., Uszkoreit J., Le Q. and Petrov S. (2019). Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* 7, 453–466.
- Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Küttler H., Lewis M., Yih W.-t., Rocktäschel T., Riedel S. and Kiela D. (2020). *Retrieval-augmented generation for knowledge-intensive nlp tasks*. In *Advances in Neural Information Processing Systems*, 33, pp. 9459–9474.
- Li J., Cheng X., Zhao X., Nie J.-Y. and Wen J.-R. (2023). *HaluEval: a large-scale hallucination evaluation benchmark for large language models*. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore. Association for Computational Linguistics, pp. 6449–6464.
- Lin D. (1998). *Automatic retrieval and clustering of similar words*. In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*.
- Lyu Y., Li Z., Niu S., Xiong F., Tang B., Wang W., Wu H., Liu H., Xu T. and Chen E. (2024). CRUD-RAG: a comprehensive chinese benchmark for retrieval-augmented generation of large language models. ArXiv preprint abs/2401.17043.
- McCarthy J. (1969). Programs with common sense. In Minsky M.L. (ed), *Semantic Information Processing*. Cambridge, MA, USA: The MIT Press, pp. 403–418.
- Meyer D., Schvaneveldt R. and Ruddy M. (1975). Loci of contextual effects on visual word-recognition. In Rabbitt P. and Dornic S. (eds), *Attention and Performance V*. New York: Academic Press.
- Nasr M., Carlini N., Hayase J., Jagielski M., Cooper A.F., Ippolito D., Choquette-Choo C.A., Wallace E., Tramèr F. and Lee K. (2023). Scalable extraction of training data from (production) language models. arXiv preprint arXiv: 2311.17035.
- Palermo D. and Jenkins J. (1964). *Word Association Norms*. Minneapolis, MN: University of Minnesota Press.
- Petroni F., Piktus A., Fan A., Lewis P., Yazdani M., De Cao N., Thorne J., Jernite Y., Karpukhin V., Maillard J., Plachouras V., Rocktäschel T. and Riedel S. (2021). *KILT: a benchmark for knowledge intensive language tasks*. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 2523–2544, Online.
- Ribeiro M.T., Singh S. and Guestrin C. (2016). *why should i trust you?": explaining the predictions of any classifier*. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144.
- Rosenthal S., Sil A., Florian R. and Roukos S. (2024). CLAPNQ: cohesive long-form answers from passages in natural questions for rag systems. ArXiv, abs/2404.02103.
- Saad-Falcon J., Khattab O., Potts C. and Zaharia M. (2023). ARES: an automated evaluation framework for retrieval-augmented generation systems. ArXiv preprint abs/2311.09476.
- Sander D.P. and Dietz L. (2021). *EXAM: how to evaluate retrieve-and-generate systems for users who do not (yet) know what they want*. In *Biennial Conference on Design of Experimental Search & Information Retrieval Systems*.
- Wei J., Wang X., Schuurmans D., Bosma M., Xia F., Chi E., Le Q.V. and Zhou D. (2022). *Chain-of-thought prompting elicits reasoning in large language models*. In *Advances in Neural Information Processing Systems*, 35, pp. 24824–24837.
- Wu Y., Zhu J., Xu S., Shum K., Niu C., Zhong R., Song J. and Zhang T. (2023). Ragtruth: a hallucination corpus for developing trustworthy retrieval-augmented language models. ArXiv preprint abs/2401.00396.
- Xiong G., Jin Q., Lu Z. and Zhang A. (2024). Benchmarking retrieval-augmented generation for medicine. ArXiv preprint abs/2402.13178.
- Yang S., Sun R. and Wan X. (2023). A new benchmark and reverse validation method for passage-level hallucination detection. In Bouamor H., Pino J. and Bali K. (eds), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore*. Association for Computational Linguistics, pp. 3898–3908.
- Zhao P., Zhang H., Yu Q., Wang Z., Geng Y., Fu F., Yang L., Zhang W. and Cui B. (2024). Retrieval-augmented generation for ai-generated content: a survey. ArXiv preprint abs/2402.19473.