Enhancing Chatbot Performance with Retrieval Augmented Generation and Prompt Engineering

Jorge Rivera Mancilla[1], Scarlett Zapata Cortés[1], Ricardo Pizarro Carreño[2] and Brian Keith Norambuena[1,*]

[1]*Universidad Católica del Norte, Antofagasta, Chile.*
[2]*Universidad de Alcalá, Alcalá de Henares (Madrid), Spain.*

Abstract

This paper presents a case study of enhancing a chatbot's performance within a Software-as-a-Service (SaaS) platform for behavioral analysis. The study focuses on the integration of Retrieval Augmented Generation (RAG) and the application of prompt engineering techniques to improve the chatbot's domain-specific knowledge and adherence to organizational guidelines. The proposed solution encompasses three key approaches: implementing an automated Extract, Transform, Load (ETL) process for efficient data updates, leveraging RAG to incorporate domain-specific information, and applying prompt engineering to ensure compliance with rules and directives. The chatbot, named Selene, is built upon the GPT-4 language model and aims to provide in-depth analysis and practical recommendations to optimize organizational behavioral development. The study follows a systematic methodology, including requirements engineering and iterative development based on the Generative AI Project Life Cycle Framework. Based on the evaluation of the chatbot using different metrics with the DeepEval library, the analysis of results from our study demonstrate the abilities of the chatbot in providing accurate and appropriate responses, incorporating domain-specific knowledge, and aligning with organizational rules. The study discusses lessons learned, provides recommendations for practitioners. In general, this work contributes to the field of conversational AI by showcasing the application of RAG and prompt engineering techniques in a domain-specific SaaS platform.

Keywords

Information Retrieval, Prompt Engineering, Large Language Models

1. Introduction

In recent years, chatbots have become increasingly prevalent in various domains, serving as virtual assistants, customer support agents, and knowledge dissemination tools [1]. The rapid advancements in artificial intelligence (AI) and natural language processing have enabled the development of sophisticated chatbots capable of engaging in human-like conversations and providing valuable assistance [2]. In the domain of behavioral analysis [3, 4], chatbots have the potential to support professionals and organizations by offering insights, recommendations, and guidance based on established principles and best practices.

WDKE'24: Workshop on Data and Knowledge Engineering, September 04, 2024, Copiapó, Chile

jorge.rivera01@alumnos.ucn.cl (J. Rivera Mancilla); scarlett.zapata@alumnos.ucn.cl (S. Zapata Cortés); ricardo.pizarroc@uah.es (R. Pizarro Carreño); brian.keith@ucn.cl (B. Keith Norambuena)
https://briankeithn.github.io/ (B. Keith Norambuena)
0000-0001-5734-8962 (B. Keith Norambuena)

Behavioral analysis is a scientific approach to understanding, predicting, and influencing human behavior in various contexts, such as organizational development, education, and healthcare [5, 6]. It involves the systematic study of observable behaviors to develop interventions that promote desired outcomes [7].

Chatbots offer on-demand support for team analysis, development, and improvement recommendations. Their effectiveness relies on accurate query understanding and response generation, while adhering to domain knowledge and organizational guidelines [8]. Recent advances in Large Language Models (LLMs) have enabled significant chatbot developments [9]. One of the key challenges in chatbot development is ensuring that the underlying language model is continuously updated with the latest information and can handle domain-specific terminology [10]. Additionally, chatbots must be designed to comply with organizational rules and directives, especially in sensitive domains such as healthcare, finance, and human resources [11]. To address these challenges, researchers have explored various techniques, including Retrieval Augmented Generation (RAG) [12] and prompt engineering [13].

RAG is a promising approach that combines the strengths of retrieval-based and generation-based methods [12]. By retrieving relevant information from external knowledge sources and augmenting the input to the language model, RAG enables chatbots to provide more accurate and informative responses [14]. On the other hand, prompt engineering involves carefully designing and optimizing the input prompts to guide the language model towards desired outputs [13], improving the quality and coherence of generated responses [15].

In this paper, we present a case study of enhancing a chatbot's performance within a Software-as-a-Service (SaaS) platform for behavioral analysis. The chatbot, named Selene, is built upon the GPT-4 language model [16] and aims to provide in-depth analysis and practical recommendations to optimize organizational behavioral development. We propose a solution that combines an automated Extract, Transform, Load (ETL) [17] process, RAG, and prompt engineering to address the challenges of outdated information and domain-specific terminology.

2. Background and Related Work

Chatbots have garnered significant attention from both academia and industry due to their human-computer interaction capabilities [1]. With the advent of deep learning and neural networks, chatbots have evolved to leverage Large Language Models (LLMs), such as GPT [18], BERT [19], and their variants [20]. These models are trained on massive amounts of textual data and can generate contextually relevant and coherent responses [8]. Researchers have explored various architectures and techniques to improve the performance of chatbots, including attention mechanisms [21], memory networks [22], and reinforcement learning [23].

RAG is an emerging paradigm that combines the strengths of retrieval-based and generation-based methods for natural language processing tasks [12]. RAG systems typically consist of a retriever that selects relevant passages from an external knowledge source and a generator that incorporates the retrieved information to produce the final output [14]. RAG has been successfully applied to a wide range of tasks, including question answering [12] and summarization [24]. By leveraging external knowledge, RAG models can provide more accurate and informative responses compared to purely generation-based approaches [12].

Prompt engineering refers to the systematic approach of designing and refining input prompts in order to effectively steer language models towards desired outputs [13]. Researchers have explored different approaches to prompt engineering, including manual prompt design [25], automated prompt search [26], and continuous prompt optimization [27]. Additionally, incorporating domain-specific knowledge and examples into prompts has been shown to improve the quality of generated responses [28].

3. Methodology

The methodology is inspired by the Generative AI Project Life Cycle Framework proposed by Fregly et al. [29], which provides a structured workflow for developing and deploying AI-powered applications. The first step in the methodology was requirements engineering [30] and analyzing the existing technological infrastructure of the SaaS platform to identify the necessary modifications and enhancements required to support the proposed solution.

3.1. Proposed Solution

Based on the requirements engineering and infrastructure analysis, a three-pronged solution is proposed to enhance the performance of the Selene chatbot. The first step is an *Automated ETL Process*, *Prompt Engineering*, and the use of *Retrieval Augmented Generation*. An overview of the proposed solution can be seen in Figure 1.

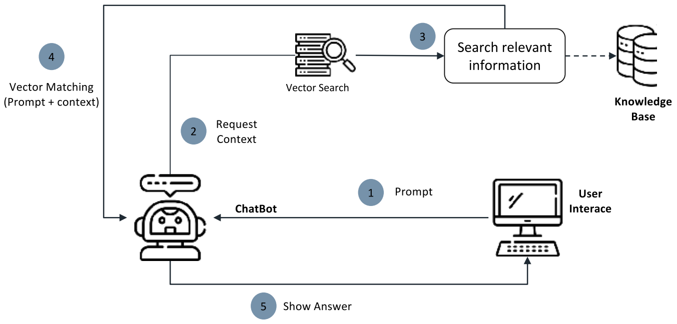

Figure 1: Overview of the Selene Chatbot within a SaaS platform for behavioral analysis.

First, an automated ETL process was developed to continuously update the chatbot's knowledge base with the most recent and domain-specific information, obtained from the company's international documentation. This process involves extracting data from various sources, transforming it into an appropriate format, and loading it into the chatbot's data storage system. The main output of this phase is the embeddings stored in the database that will be used in the subsequent RAG phase. Next, prompt engineering techniques such as zero-shot, one-shot, and few-shot prompting were applied to align the chatbot's responses with organizational rules and guidelines. Finally, RAG was employed to allow the chatbot to retrieve information from an external knowledge base by searching for relevant information through the previously mentioned embeddings for incorporation into the generated responses.

3.2. Prompt Engineering Iterations

Prompt engineering techniques have been applied to align the chatbot's responses with organizational rules and guidelines. In particular, a set of prompts, incorporating zero-shot, one-shot, and few-shot examples, has been designed. We develop multiple prompts to address a diverse range of user queries. The results of this prompt engineering process have been reviewed by domain experts. Example prompts from the system are shown in Table 3.2.

Prompt	Prompt Text
P1	"Your name is Selene, you are a virtual assistant..."
P2	"The probability of safe working conditions is a decimal number between 0 and 1. The levels are: ..."
P3	"The competencies for supervisory workers are: ..."
P4	"The function ... allows you to obtain information about the workers in the following way: ..."
P5	"The function ... allows you to consult the administration of ..."
P6	"For the response of the function ..., you must indicate where the information is located"

Table 1
Example prompts used to evaluate the chatbot (modified to preserve confidentiality).

Several prompt design patterns were applied to improve the interaction with the model [31]. For example, the *Persona Pattern*, which involves assigning a distinct role to the model, thereby facilitating the personalization and contextualization of the responses it generates [32]. Next, the *Template Pattern*, which provides a structured framework or schema for various types of queries. By utilizing predefined templates, the model can generate responses that are not only coherent but also well-structured and aligned with the query's intent [33]. We also used *Context Manager Pattern*, which is central to maintaining coherence and continuity within conversations is the context manager. This component is responsible for retaining and managing the context throughout the interaction [34]. Finally, we also considered *Input Semantics Patterns*, which establish specific rules and formats for input data. By defining clear guidelines, these patterns ensure that the model processes information accurately and efficiently [35].

3.3. Evaluation

3.3.1. Evaluation Process

We chose to evaluate our models with the *DeepEval* [36] library due to its wide range of metrics to evaluate the performance of the chatbot, including aspects of precision, coherence, hallucination, and relevance. Subsequently, two datasets were created, each containing input and expected_output fields, representing the question asked and the anticipated answer, respectively. The first dataset focused on questions designed to assess RAG, while the second dataset was used to evaluate the quality of the responses. Subsequently, use cases were defined to enable the utilization of evaluation metrics for conducting unit tests on LLM applications. Furthermore, the test data (23 cases) also had context fields to evaluate RAG-related results. We show examples of queries used throughout the evaluation process in Table 3.3.1.

Test Cases	Example Expected Response
T1: Help with area management	"To manage the company's areas, use the sidebar and follow these steps: ..."
T3: Safe work analysis	"There are ... workers who have a safe work probability at level ..."
T4: Competence analysis	"There are ... workers who have competence ... at level ..."
T5: Safest worker evaluation	"The worker with the highest safe work probability is ..."
T6: Most developed competence	"The most developed competence for workers is ..."
T7: Specific worker query	"The information of the worker with ID number ... is as follows: ..."
T8: Worker evaluation history	"The evaluation history of the worker ... is: ..."

Table 2
Example queries used to evaluate the chatbot (modified to preserve confidentiality).

3.3.2. Evaluation Metrics

Eight metrics were used to evaluate the responses generated by the LLM. First, we consider general evaluation metrics that do not involve the use of RAG and the additional context it provides but focus on general properties of the response. Answer Relevancy, which measures the quality and relevance of the generated response in relation to the question asked, ensuring pertinent and useful answers [37, 38]. *Bias*, which indicates if the chatbot's responses contain gender, racial, political, or other biases, ensuring impartial and fair responses [39]. Finally, *Toxicity*, which evaluates if the chatbot's responses contain toxic elements, such as mockery, hatred, disdainful statements, or threats, maintaining a safe and respectful environment [39].

Next, we consider metrics that seek to evaluate the responses focusing on RAG. *Faithfulness*, which evaluates whether the generated response is faithful to the retrieval context, ensuring accurate and reliable information [40, 38]. *Contextual Precision*, which measures the contextual precision of each node in the retrieval context for the question asked based on the expected response, ensuring the chatbot uses the most relevant information from the context [38]. *Contextual Recall*, which evaluates the model's ability to retrieve information aligned with the expected response and the retrieval context, preventing the omission of important information [38]. *Contextual Relevancy*, which measures the relevance of the information in the retrieval context according to the question asked, ensuring coherent responses [38]. Finally, *Hallucination*, which detects if the model generates incorrect or fabricated information by comparing it with the actual response and the provided context, preventing misleading responses [41].

4. Results and Discussion

4.1. Evaluation Results

We present the results for all 23 test cases in Figure 2. The implementation of RAG has shown promising results in enhancing the chatbot's ability to provide accurate and context-specific responses. We note that throughout these test cases, the external knowledge base has been populated with domain-specific information in the ETL phase, and the retrieval mechanism has been fine-tuned to select the most relevant passages based on user queries.

In terms of the general evaluation metrics, the answers present no issues with Bias or Toxicity. In terms of Faithfulness, the model also generally presents no issues (except in T15). In contrast, Answer Relevancy presents some slightly mixed results, but in general the models provide a decent performance in this metric.

Regarding the RAG-focused metrics, we note that contextual precision presents the worst results of all the metrics over the full range of test cases. In contrast, contextual recall is generally higher and presents a better overall performance. Contextual relevancy is considered high in 16 test cases, but severely fails in 7 instances. For the Contextual Retrieval metric, we highlight test cases T20 and T21, as they are the only ones where Contextual Precision is significantly better. In these cases, the retrieval context did not contain the specific evaluations or competency level information required to answer the queries. Upon comparing the expected and actual responses for these test cases, it was observed that the main difference was the language model's tendency to generate more extensive responses, while the core content remained the same, thus making it difficult to generate the correct answer. Finally, in terms of hallucinations, only two test cases presented issues (T17 and T20). These issues were caused by the inability of the chatbot to derive the response directly from the given context or its failure to provide requested information despite correctly identifying the worker and company in the internal functions.

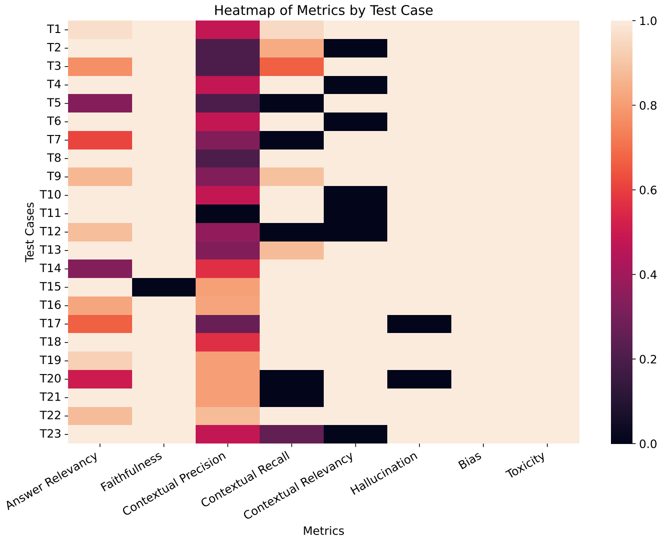

Figure 2: Heatmap showing the distribution of quality metrics for all test cases.

4.2. Discussion and Recommendations

The selection of appropriate techniques, such as RAG and prompt engineering, plays a crucial role in enhancing chatbot performance and preventing hallucinations. RAG has proven to be effective in incorporating domain-specific knowledge and providing context-specific responses. Prompt engineering techniques, such as zero-shot, one-shot, and few-shot prompting, help in guiding the chatbot's behavior and tone.

In this context, integrating domain-specific knowledge is essential for chatbots operating in specialized domains, such as behavioral analysis. Practitioners should invest in building comprehensive knowledge bases that cover relevant domain-specific information. This can be achieved through collaboration with domain experts, extraction of information from existing documentation, and continuous updates based on new findings and research. The use of RAG can help in leveraging external knowledge sources effectively.

Finally, despite the capabilities of the model in generating mostly appropriate responses, there are certain limitations to the current implementation that need to be addressed in future work. One limitation is the reliance on a single external knowledge base for RAG, which may not cover all the relevant domain-specific information. Furthermore, future work should prioritize qualitative analysis involving actual users and behavioral analysis experts.

5. Conclusions

In this paper, we presented a case study of enhancing the performance of the Selene chatbot within a SaaS platform for behavioral analysis. The proposed solution incorporates an automated ETL process, Retrieval Augmented Generation, and prompt engineering techniques to address the challenges of outdated information, domain-specific terminology, and adherence to organizational guidelines. The analysis of results demonstrates the effectiveness of the enhanced chatbot in improving response accuracy and addressing these challenges. Thus, the enhanced Selene chatbot serves as a promising example of how RAG and prompt engineering can be leveraged to improve the performance of conversational AI systems. Future work could include fine-tuning the underlying language model to handle the domain of behavioral analysis. Additionally, benchmarking the Selene chatbot against other LLMs could provide valuable insights into its relative performance. Finally, implementing a continuous feedback mechanism that allows end users to directly rate and comment on chatbot responses could also be a valuable enhancement that could further refine the chatbot's capabilities.

Acknowledgments

We would like to thank Pignus for their support. This work was partially supported by the project 2023-11010033-VRIDT-UCN.

References

[1] A. Følstad, P. B. Brandtzæg, Chatbots and the new world of hci, interactions 24 (2017) 38–42.

[2] R. Dale, The return of the chatbots, Natural language engineering 22 (2016) 811–817.

[3] Y. Y. Chiu, A. Sharma, I. W. Lin, T. Althoff, A computational framework for behavioral assessment of llm therapists, arXiv preprint arXiv:2401.00820 (2024).

[4] A. Aggarwal, C. C. Tam, D. Wu, X. Li, S. Qiao, Artificial intelligence–based chatbots for promoting health behavioral changes: systematic review, Journal of medical Internet research 25 (2023) e40789.

[5] J. O. Cooper, T. E. Heron, W. L. Heward, et al., Applied behavior analysis (2007).

[6] W. W. Fisher, C. C. Piazza, H. S. Roane, Handbook of applied behavior analysis, Guilford Publications, 2021.

[7] G. R. Mayer, B. Sulzer-Azaroff, M. Wallace, Behavior analysis for lasting change, Sloan Pub., 2012.

[8] J. Gao, M. Galley, L. Li, Neural approaches to conversational ai, in: The 41st intl. ACM SIGIR conf. on research & development in information retrieval, 2018, pp. 1371–1374.

[9] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili, et al., A survey on large language models: Applications, challenges, limitations, and practical usage, Authorea Preprints (2023).

[10] H. Chen, X. Liu, D. Yin, J. Tang, A survey on dialogue systems: Recent advances and new frontiers, Acm Sigkdd Explorations Newsletter 19 (2017) 25–35.

[11] A. Miner, A. Chow, S. Adler, I. Zaitsev, P. Tero, A. Darcy, A. Paepcke, Conversational agents and mental health: Theory-informed assessment of language and affect, in: Proc. of the fourth international conference on human agent interaction, 2016, pp. 123–130.

[12] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in Neural Information Processing Systems 33 (2020) 9459–9474.

[13] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, ACM Computing Surveys 55 (2023) 1–35.

[14] G. Izacard, E. Grave, Leveraging passage retrieval with generative models for open domain question answering, arXiv preprint arXiv:2007.01282 (2020).

[15] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners, arXiv preprint arXiv:2012.15723 (2020).

[16] M. S. Rahaman, M. T. Ahsan, N. Anjum, H. J. R. Terano, M. M. Rahman, From chatgpt-3 to gpt-4: a significant advancement in ai-driven nlp tools, Journal of Engineering and Emerging Technologies 2 (2023) 1–11.

[17] J. C. Nwokeji, R. Matovu, A systematic literature review on big data extraction, transformation and loading (etl), in: Intelligent Computing: Proc. of the 2021 Computing Conference, Volume 2, Springer, 2021, pp. 308–324.

[18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.

[19] J. D. M.-W. C. Kenton, L. K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proc. of NAACL-HLT, 2019, pp. 4171–4186.

[20] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, in: International Conference on Learning Representations, 2019.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[22] S. Sukhbaatar, J. Weston, R. Fergus, et al., End-to-end memory networks, Advances in neural information processing systems 28 (2015).

[23] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, D. Jurafsky, Deep reinforcement learning for dialogue generation, arXiv preprint arXiv:1606.01541 (2016).

[24] A. Fan, C. Gardent, C. Braud, A. Bordes, Using local knowledge graph construction to scale seq2seq models to multi-document inputs, in: 2019 Conf. on Empirical Methods in

Natural Language Processing and 9th Intl. Joint Conf. on NLP, 2019.

[25] L. Reynolds, K. McDonell, Prompt programming for large language models: Beyond the few-shot paradigm, in: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–7.

[26] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, S. Singh, Autoprompt: Eliciting knowledge from language models with automatically generated prompts, in: Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 4222–4235.

[27] Z. Xu, C. Wang, M. Qiu, F. Luo, R. Xu, S. Huang, J. Huang, Making pre-trained language models end-to-end few-shot learners with contrastive prompt tuning, in: Proc. of the Sixteenth ACM Intl. Conf. on Web Search and Data Mining, 2023, pp. 438–446.

[28] J. Wei, M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, Q. V. Le, Finetuned language models are zero-shot learners, in: International Conference on Learning Representations, 2021.

[29] C. Fregly, A. Barth, S. Eigenbrode, Generative AI on AWS: building context-aware multimodal reasoning applicaions, O'Reilly Media, 2023.

[30] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, T. Zimmermann, Software engineering for machine learning: A case study, in: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), IEEE, 2019, pp. 291–300.

[31] D. C. Schmidt, J. Spencer-Smith, Q. Fu, J. White, Cataloging prompt patterns to enhance the discipline of prompt engineering, URL: https://www. dre. vanderbilt. edu/~schmidt/PDF/ADA_Europe_Position_Paper. pdf [accessed 2023-09-25] (2023).

[32] G. Sun, N. Zhan, J. Such, Building better ai agents: A provocation on the utilisation of persona in llm-based conversational agents, in: Proc. of the 6th International Conference on Conversational User Interfaces, CUI 2024, 2024.

[33] M. Fromm, M. Fromm, Design of llm prompts for iterative data exploration (2023).

[34] S. de Kinderen, K. Winter, Towards taming large language models with prompt templates for legal grl modeling, in: International Conference on Business Process Modeling, Development and Support, Springer, 2024, pp. 213–228.

[35] J. Santen, Using LLM Chatbots to Improve the Learning Experience in Functional Programming Courses, B.S. thesis, University of Twente, 2024.

[36] T. C. AI, Deepeval: A benchmarking framework for language learning models, URL: https:github.com/confident-ai/deepeval (2023).

[37] M. Desai, R. G. Mehta, D. P. Rana, A model to identify redundancy and relevancy in question-answer systems of digital scholarly platforms, Procedia Computer Science 218 (2023) 2383–2391.

[38] K. Juvekar, A. Purwar, Cos-mix: Cosine similarity and distance fusion for improved information retrieval, arXiv preprint arXiv:2406.00638 (2024).

[39] Y. P. Chetnani, Evaluating the Impact of Model Size on Toxicity and Stereotyping in Generative LLM, Ph.D. thesis, State University of New York at Buffalo, 2023.

[40] J. Maynez, S. Narayan, B. Bohnet, R. McDonald, On faithfulness and factuality in abstractive summarization, arXiv preprint arXiv:2005.00661 (2020).

[41] V. Rawte, S. Tonmoy, K. Rajbangshi, S. Nag, A. Chadha, A. P. Sheth, A. Das, Factoid: Factual entailment for hallucination detection, arXiv preprint arXiv:2403.19113 (2024).