


Measuring the Feasibility of a Question and Answering System for the Sarawak Gazette Using Chatbot Technology

Yasir Lutfan bin Yusuf, Suhaila binti Saeed 

Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Kota Samarahan, Sarawak, Malaysia

Corresponding author: Yasir Lutfan bin Yusuf (rtendt1@gmail.com)

Editorial Record

First submission received:
November 20, 2024

Revision received:
March 5, 2025

Accepted for publication:
March 6, 2025

Academic Editor:
Adela Jarolimkova
Charles University, Czech Republic

This article was accepted for publication
by the Academic Editor upon evaluation of
the reviewers' comments.

How to cite this article:
Yusuf, Y. L. Y., & Saeed, S. (2025).
Measuring the Feasibility of a Question and
Answering System for the Sarawak Gazette
Using Chatbot Technology. *Acta
Informatica Pragensia*, 14(3), 365–392.
<https://doi.org/10.18267/j.aip.263>

Copyright:
© 2025 by the author(s). Licensee Prague
University of Economics and Business,
Czech Republic. This article is an open
access article distributed under the terms
and conditions of the [Creative Commons
Attribution License \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/).



Abstract

Background: The Sarawak Gazette is a critical repository of information pertaining to Sarawak's history. It has received much attention over the last two decades, with prior studies focusing on digitizing and extracting the gazette's ontologies to increase the gazette's accessibility. However, the creation of a question answering system for the Sarawak Gazette, another avenue that could improve accessibility, has been overlooked.

Objective: This study created a new system to generate answers for user questions related to the gazette using chatbot technology.

Methods: This system sends user queries to a context retrieval system, then generates an answer from the retrieved contexts using a Large Language Model. A question answering dataset was also created using a Large Language Model to evaluate this system, with dataset quality assessed by 10 annotators.

Results: The system achieved 55% higher precision, and 42% higher recall compared to previous state-of-the-art historical document question answering while only sacrificing 11% of cosine similarity. The annotators overall rated the dataset 2.9 out of 3.

Conclusion: The system could answer the general public's questions about the Sarawak Gazette in a more direct and friendly manner compared to traditional information retrieval methods. The methods developed in this study are also applicable to other Malaysian historical texts that are written in English. All code used in this study have been released on GitHub.

Index Terms

Historical documents; Old newspapers; Accessibility; Question answering; Artificial intelligence; Retrieval augmented generation; LangChain.

1 INTRODUCTION

The Sarawak Gazette is recognized as being one of the most important repositories in learning the history of Sarawak (Rosita et al, 2010). This gazette contains invaluable information on topics ranging from politics to people's living standards, government, the landscape and even flora and fauna. The main goal of Sarawak Gazette was to report on Sarawak's native relations, trading activities, the conditions of the government's various residences and other public interests for international distribution (Rosita et al, 2010). These papers were published between August 1870 to January 1984 (Rosita et al, 2010; Fong & Ranaivo-Malançon, 2015), before computers became commonplace.

Unfortunately, it is difficult to search for the required information as the documents need to be manually searched and read thoroughly (Rosita et al, 2010). Because of their age, they have become quite fragile and must also be handled with the utmost care.

Moreover, the original papers of the Sarawak Gazette have been scattered across Sarawak's various libraries such as the Sarawak Museum and the Sarawak State Library (Pustaka Sarawak), as well as other countries' libraries such as the National Library of Australia (Rosita et al, 2010). Thankfully, there have been efforts made at digitizing the Sarawak Gazette such that they are now widely accessible to the public (Rosita et al, 2010). There have also been manual efforts made at dealing with spelling changes, obsolete names, and grammatical errors, as well as in providing extra details within the digitized version of Sarawak Gazette (Rosita et al, 2010).

While digital libraries such as Pustaka Negeri Sarawak State Library do not even have searching capabilities as of writing (Pustaka Negeri Sarawak, 2013), the gazette overall has been digitized and cleaned up enough such that there have been serious efforts made at extracting the ontologies of such documents (Rosita et al, 2010; Ramli et al, 2017). Ontology is a shareable conceptualization of a domain that is specified formally and explicitly, to facilitate knowledge sharing between systems (Taye, 2010). If successful, these ontologies can allow semantic web technologies to be implemented, making the gazette machine-understandable rather than just machine-readable (Taye, 2010). For example, search engines could more effectively perform the task of Information Retrieval (IR) on Sarawak Gazette by using the ontology as a thesaurus to also match words that are similar to the query's words (Paolucci, Kawamura, Payne, and Sycara, 2002; Taye, 2010). Information Retrieval here refers to the task of retrieving a sorted list of documents that are ranked based on a user's query, with resulting IR systems commonly referred to as search engines (Jurafsky & Martin, 2024).

Although there has been work done in trying to implement IR technology for the Sarawak Gazette, virtually no attention seems to have been given to another possible aspect for increasing the accessibility of the Sarawak Gazette, which is by implementing a Question Answering System (QAS). A QAS is any system that can answer a user's question, whether through manual effort such as Community Question Answering sites (CQAs) or through automated means such as language models. Unlike Information Retrieval (IR) systems, QASes are expected to directly give short answers to user questions instead of returning a ranked list of documents containing the answers (Allam & Haggag, 2012; Ojokoh & Adebisi, 2019).

QASes are typically implemented with the retriever/reader architecture, utilizing two main components: the retrieval component and the reader component (Jurafsky & Martin, 2024). The retriever component uses IR to get a set of documents from a user's question, while the reader component parses the set of documents retrieved by IR to arrive at a definitive answer. The reader component itself can either return a span of text in the documents as the answer or generate novel sentences containing the answer. The latter method specifically is known as Retrieval Augmented Generation (RAG). The main advantage of the retriever/reader architecture is the ability to ask questions relating to proprietary or internal data not previously encountered by large language models during their training (Jurafsky & Martin, 2024).

Building a QAS has become significantly more feasible due to the recent emergence of libraries such as LangChain that allow for simple integration of chatbots into new applications (Introduction, 2024). Chatbots are computer systems that can simulate human conversation, usually over the Internet (Adamopoulou & Moussiades, 2020). They can perform a variety of tasks including education, information retrieval as well as question answering (Adamopoulou & Moussiades, 2020). Chatbots and QASes are not the same – chatbots are built with human-computer interactivity as the main principle and are meant to emulate the entire spectrum of informal human conversation including question answering (Jurafsky & Martin, 2024), while QASes are meant to perform question answering only – QASes do not necessarily have to be able to hold a conversation.

There are four distinct techniques for applying chatbot technology in building a QAS, which are prompt engineering, retrieval augmented generation (RAG), fine-tuning and pre-training. Each of these vary in the amount of control they offer – prompt engineering and retrieval augmented generation do not tamper with the model's internal parameters, while fine-tuning makes less parameter changes than pre-training (Ling et al, 2023). These techniques can be sorted by data required, where prompt engineering requires the least amount of prepared data, followed by RAG, fine-tuning and lastly pre-training, which requires the most data. There is also a positive correlation between model performance and computational power required – prompt engineering and RAG are typically less demanding, but also typically do not match performance levels reached by fine-tuning and pre-training (Ling et al, 2023).

Despite the simplicity of prompt engineering, it may not be effective for building a QAS for the Sarawak Gazette. While there has been a lot of training data put into chatbots, chatbots almost never contain data on private, recent, or otherwise obscure corpora (Tebbe, 2024), such as the Sarawak Gazette. Using RAG instead remedies this – by supplying a question as well as the appropriate context, chatbots will still be able to pluck the correct answer from the given context despite not ever encountering them during training (Tebbe, 2024).

Testing the QAS is another key step as it reveals how well the system answers the questions given to it. However, a fundamental issue of testing the QAS is that this requires a dataset of questions and reference answers that can be answered from reading the Sarawak Gazette, which has also not been addressed by previous research. To address this, the gazette's contents is used to automatically generate a reference question-answering dataset. This dataset is then evaluated with human annotators to ensure all questions and reference answers are accurate.

As the authors of this study anticipate future research by others to follow-up on implementing semantic web technology for the Sarawak Gazette, this study instead aims to understand how a QAS can be implemented that uses the context of Sarawak Gazette and using chatbot technology to directly give users the answers they need. This can provide an alternative path in increasing the accessibility of the Sarawak Gazette for historians as well as the public eye, as well as provide insight for future researchers on potential, additional challenges when it comes to raising the gazette's accessibility.

In summary, the Sarawak Gazette has remained quite inaccessible due to its age, but efforts at digitization and in making these documents searchable have been made. However, there has yet to be any research done on another path of improving the accessibility of these documents, which is by implementing a QAS using RAG and chatbot technology that can be used by the public.

The rest of this paper is organized as follows: Section 1.1 states the problem statement, research questions, objectives and outcomes. Section 1.2 reviews the literature while Section 2 elaborates this study's methodology. Section 3 presents the results of this study, then Section 3.1 analyses these results. Finally, Section 4 and Section 4.1 discuss the obtained results and state the authors' conclusions, respectively.

1.1 Problem Statement

Efforts in digitization have been made to also allow old newspapers such as the Sarawak Gazette to be more accessible for historians (Bingham, 2010). But even after such efforts, challenges still arise when trying to directly search for specific information, as techniques such as keyword searching will give a set of articles rather than the information directly (Bingham, 2010). As there are no existing systems that support advanced queries on the Sarawak Gazette that could solve the issue, manual interpretation of the articles is needed to get the required information, which is laborious and time-consuming. This can cause researchers, historians and the public to make assumptions grounded on investigations that are superficial and hurried, resulting in ineffective utilization of the potentially wealthy contents of old newspapers (Bingham, 2010).

Even after the Sarawak Gazette's accessibility has been improved after being digitized, barriers remain in handling the historical data's variability and inconsistencies. These not only include problems such as grammatical errors, but also the use of outdated names, words and spellings that have changed since the gazette was printed. This hampers the effectiveness of both Information Retrieval and Question Answering systems. The current approaches for addressing these inconsistencies are mostly manual and tedious, which can lead to inefficiencies and even inaccuracies. The lack of a systematic method for gathering the necessary generalizations of the historical data contained in the Sarawak Gazette further complicates the production of an accurate and robust Question Answering System (QAS).

1.1.1 Research Questions

This paper is driven by the following two research questions:

1. What is the accuracy of a QAS that utilizes chatbot technology in answering questions related to the Sarawak Gazette?
2. What is the quality of a Sarawak Gazette QAS testing dataset automatically generated using a Large Language Model?

1.1.2 Research Objectives

The primary objective of this study is to implement a QAS for the Sarawak Gazette that will provide accurate and direct answers to user questions using chatbot technology. This includes leveraging retrieval augmented generation (RAG) to allow the chatbot to use the gazette's rich contents for generating contextually relevant answers. The performance of the resulting QAS is then evaluated against the previous state-of-the-art. To limit this project's scope, users can only give questions in text form, and answers can also only be given in text form.

The secondary objective of this study is to develop and implement a methodology for processing the Sarawak Gazette's historical data that can deal with various inconsistency problems including outdated spellings, obsolete names, and grammatical errors. This methodology aims to improve the accuracy and reliability of a QAS for the Sarawak Gazette by refining the QAS's ability to understand the historical data and generate responses that are contextually relevant. This leads to increased usability and accessibility of the Sarawak Gazette's rich historical content through the question answering system.

1.1.3 Research Outcomes

The main deliverable of this research is a Python program that allows the user to input their question and receive their answer without needing to provide the context behind the question, as long as the question's context is within the Sarawak Gazette. The answers given by the program are then evaluated with automated metrics and discussed in this report.

This study will also use the Sarawak Gazette to automatically generate a reference question answering dataset, then evaluate the resulting dataset using annotator-assigned evaluation scores. A discussion will be held about the quality of this study's testing dataset.

1.2 Literature Review

This section will review previous research related to the creation of a question answering system (QAS) for Sarawak Gazette using chatbot technology. This section is composed of eight subsections, which in order talk about research done on the accessibility of Sarawak Gazette, the importance, evolution, models, techniques, and evaluation of QASes, the use of QASes over historical texts and lastly a discussion of findings.

1.2.1 The Sarawak Gazette

Little research has been found on enhancing the accessibility of the Sarawak Gazette. This subsection will talk about the contributions of all four research papers that have studied this topic.

The Sarawak Gazette was a monthly newspaper first published in 1870 and today is regarded as an important repository of Sarawak's history, containing information ranging from politics, living standards and conditions, law and order, ethnic relations, Sarawak's landscape, economy, flora, and fauna (Rosita et al, 2010; Fong & Ranaivo-Malançon, 2015). Rosita et al (2010), Fong and Ranaivo-Malançon (2015) note that due to the age of these papers, some have significantly deteriorated and require delicate handling, and some others are physically scattered across different libraries. Because of these issues, there have been efforts made at digitizing such documents. Figures 1, 2 and 3 show the scanned and digitized versions of the Sarawak Gazette.

The digitization process involved the usage of an Optical Character Recognition (OCR) machine to extract the gazette's texts and store them into plaintext files, but this process was far from error-free with only 60% of characters being recognizable (Rosita et al, 2010). Figure 4 shows an example of these errors. Manual efforts have also been made at dealing with spelling changes, obsolete names, and grammatical errors (Rosita et al, 2010). Other research on this topic aimed to extract metadata about the gazette's structure to reduce the number of human corrections required, which has yielded six Text Encoding Initiative (TEI) XML templates to represent the different layouts of the gazette's pages (Fong & Ranaivo-Malançon, 2015).

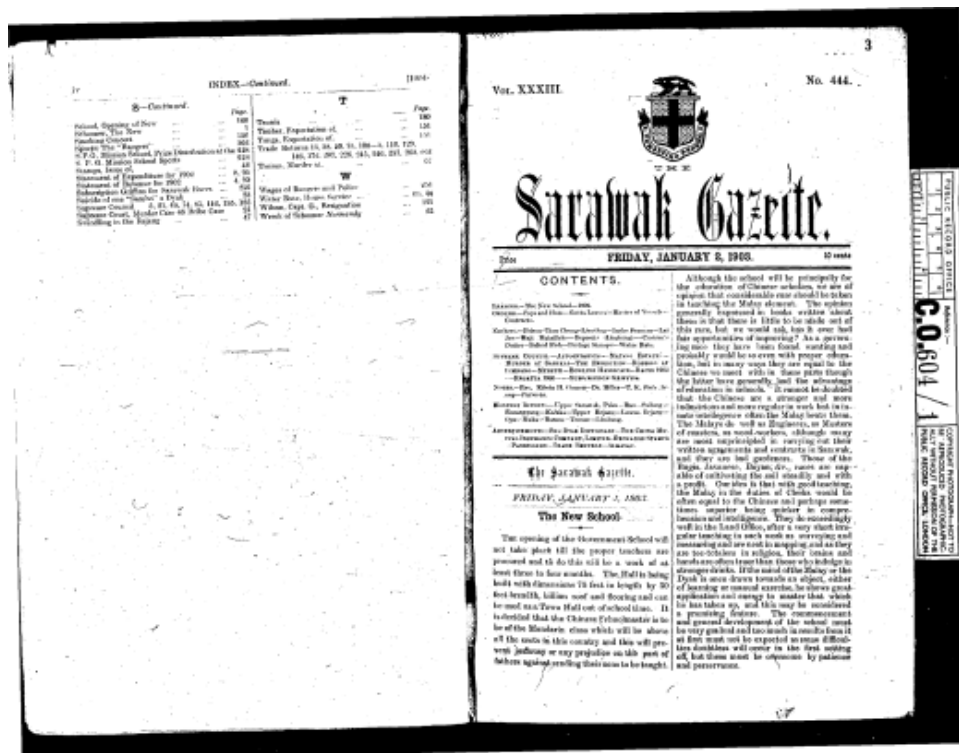


Figure 1. A scan of the Sarawak Gazette. Source: (Fong & Ranaivo-Malançon, 2015).

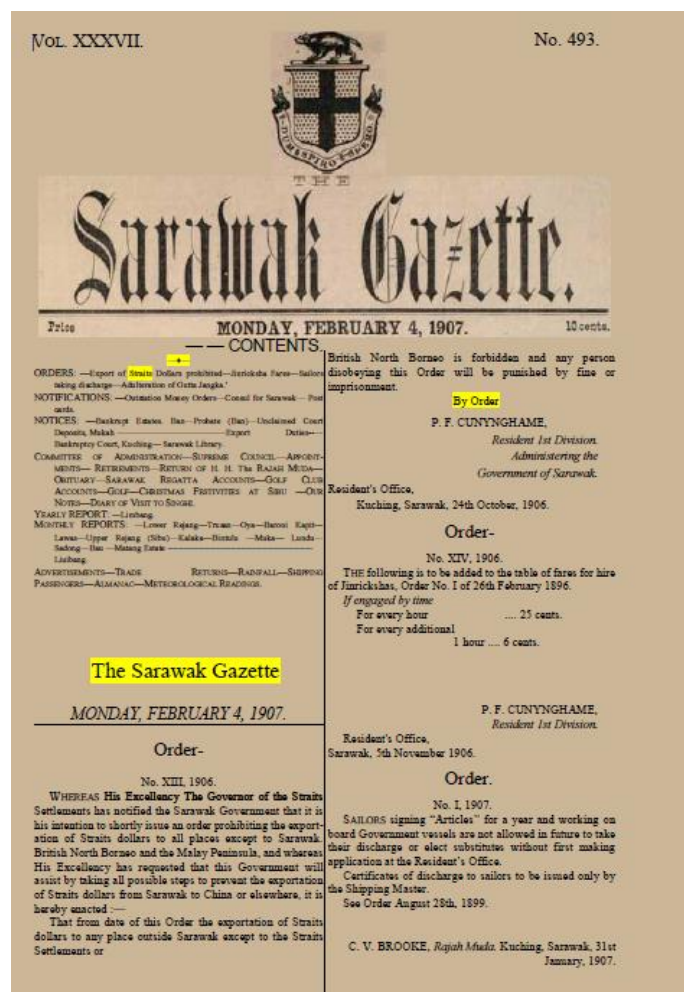


Figure 2. A screenshot of issue 493 of the digitized Sarawak Gazette, in PDF format.

VOL. XXXVII.
 No. 493.
 — CONTENTS.
 — ? —
 ORDERS: —Export of Straits Dollars prohibited—Jinricksha Fares—Sailors taking discharge—Adulteration of Gutta Jangka.
 NOTIFICATIONS: —Outstation Money Orders—Consul for Sarawak—Post cards.
 NOTICES: —Bankrupt Estates. Ban—Probate (Ban)—Unclaimed Court Deposits, Mukah Export Duties—Bankruptcy Court, Kuching—Sarawak Library.
 COMMITTEE OF ADMINISTRATION—SUPREME COUNCIL—APPOINTMENTS—RETIREMENTS—RETURN OF H. H. The RAJAH MUDA—OBITUARY—SARAWAK REGATTA ACCOUNTS—GOLF CLUB ACCOUNTS—GOLF—CHRISTMAS FESTIVITIES AT SIBU—OUR NOTES—DIARY OF VISIT TO SINGHI.
 YEARLY REPORT: —Limbang.
 MONTHLY REPORTS: —Lower Rejang—Trusan—Oya—Baroni Kapit—Lawas—Upper Rejang (Sibu)—Kalaka—Bintulu—Muka—Lundu—Sadong—Bau—Matang Estate Liuibang.
 ADVERTISEMENTS—TRADE RETURNS—RAINFALL—SHIPPING PASSENGERS—ALMANAC—METEOROLOGICAL READINGS.
 The Sarawak Gazette
 MONDAY, FEBRUARY 4, 1907.
 Order-
 No. XIII, 1906.
 WHEREAS His Excellency The Governor of the Straits Settlements has notified the Sarawak Government that it is his intention to shortly issue an order prohibiting the exportation of Straits dollars to all places except to Sarawak, British North Borneo and the Malay Peninsula, and whereas His Excellency has requested that this Government will assist by taking all possible steps to prevent the exportation of Straits dollars from Sarawak to China or elsewhere, it is hereby enacted :—
 That from date of this Order the exportation of Straits dollars to any place outside Sarawak except to the Straits Settlements or

Figure 3. An excerpt of issue 493 of the digitized Sarawak Gazette, in raw text format.

CONTENTS.
 Leaders. - The New School - 1902.
 Orders. - Caps and Hats - Gutta Leaves - Master of Vessels - Contracts.
 Notices. - Prison-Tian Chung-Lim Ong - Inche Draman - Lai Jee - Haji —
 Deposit* (Lim\>a'n<); —Custom's
 Duties—Salted Fish—Postage Stump* —Water Kate.
 Hipkeme Council—Appointments—Matar<; Estate—murder at
 sarikai—tle expedition'—ropbixg at Limbano—Museum—Bowlisq
 Handicaps—Races 1903 —Regatta 1903—Subscription Griffins.
 Notes.—Rev. Edwin H. flomei—Dr. Hiller—T. K. Itatn .lr-ting—Patricia.
 Moxtiily Rf.kh-ts.—Uppci Sarawak, Paku—B&u—Sailonj; —
 Simanssang—Kalaku—Upper Rejanx —l.o»-«. liejan^ — O.va—Muka—
 Baram—Trusan—Limbang.
 Advertisements—Sea Dtak Dictioxart —The China Mutual Insurance
 Company. Limited-Exchange Stamps —Passengebs—Trade Returns—
 Almanac.

Figure 4. Example OCR errors. Source: (Fong & Ranaivo-Malançon, 2015).

There also exists research aiming to extract the ontologies contained in the gazette using the Karlsruhe Ontology and Semantic Web (KAON) framework, creating a document in XML format containing data required by the semantic web (Rosita et al, 2010). STOLE and Simple News and Press Ontologies (SNaP) were also used in another study for the same purpose (Ramli et al, 2017). Other than that, separate research also focused on understanding popular topics in the Sarawak Gazette that were still shared on social media (Nor Azizan et al, 2023). The authors of that study did so by performing analysis on a corpus of tweets on Twitter related to the gazette using Latent Dirichlet Allocation (LDA), followed by using a Convolutional Neural Network (CNN) to further classify the topics.

In summary, there have been efforts made in digitizing the Sarawak Gazette and in making the gazette documents searchable. This implies that enough documents have been digitized such that the ontologies of the Sarawak Gazette can be meaningfully extracted.

1.2.2 The Need for Question and Answering Systems

This subsection will review the concept of QASes, the difference of these systems compared to information retrieval systems and ideas on the construction of such systems.

QASes aim to automatically provide the best answers to queries expressed by the user in natural language (Hovy, Gerber, Hermjakob, Junk & Lin, 2000; Dumais et al, 2002; Ong, Day & Hsu, 2009; Ojokoh & Adebisi, 2019). Unlike Information Retrieval (IR) systems, QASes are expected to directly give the answers to questions instead of returning a ranked list of documents containing the answers (Allam & Haggag, 2012; Ojokoh & Adebisi, 2019). Some

researchers even argue that IR systems such as search engines do not truly perform the task of information retrieval as time is consumed in examining every retrieved document one by one to actually get the desired information (Ferret et al, 2001; Mishra & Jain, 2015). QASes can either be open-domain, which deal with almost every domain, or closed-domain, which are specialized to work under a specific domain such as music or weather forecasting (Allam & Haggag, 2012).

This is a challenging task due to the several types of questions that users can ask, such as definitional questions, listing questions and why-type questions (Allam & Haggag, 2012). However, the main type of questions submitted by users are factoid questions – questions which are unanswerable such as the year of the Egyptian revolution (Allam & Haggag, 2012). Another problem is that different users can ask the same question but phrase it differently, which would require query expansion for such questions to be resolved (Allam & Haggag, 2012).

In summary, the purpose of QASes is to automatically serve a concise, direct answer from a user-input question, without having the user waste time in searching through the answer. But building a question answer system is a challenging task due to the variety of questions that could be posed.

1.2.3 Evolution of Question Answering Systems

This subsection will discuss technologies created for answering user questions, starting with community question answering sites and reviewing them up to ChatGPT, then review GPT4All's contributions.

One form of question answering system that existed before chatbot technology had matured is Community Question Answering sites (CQAs) (Roy et al, 2022). These sites are on the world-wide web and allow users to exchange questions and answers with other users. One example of a popular community question answering site that remains to the present day is Stack Overflow (Roy et al, 2022).

CQAs allow users to give questions and answers simply with natural language, but these systems also have major drawbacks. One drawback that is of particular importance to this paper is the amount of time it takes to receive an answer to a question. Any user that posts a question must wait to receive their answers, as the right user must have seen the question and must also post the right answer for it, both of which can lead to questions not receiving any answers for a long time, if not forever (Roy et al, 2022). Other issues relating to CQAs are shown in Figure 5.

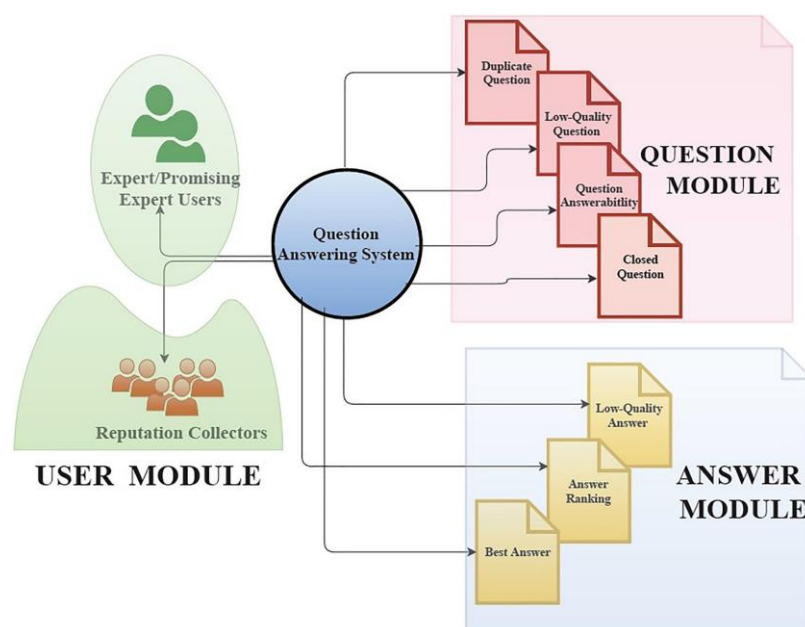


Figure 5. Review of issues in community question answering systems. Source: (Roy et al, 2022).

In 1950, the idea of chatbots was popularized after Alan Turing (1950) proposed the Turing Test for evaluating how human a machine behaves (Adamopoulou & Moussiades, 2020). The first known instance of a chatbot was ELIZA, published in a paper by Joseph Weizenbaum (1966) that aimed to act as a psychotherapist simply by transforming the user's prompts into question form. Its conversational abilities were weak, but it was good enough to confuse people at the time (Adamopoulou & Moussiades, 2020). An improvement over ELIZA was the PARRY chatbot,

published in 1971 (Colby et al, 1971). Unlike ELIZA, this chatbot was meant to emulate a paranoid person and was able to express feelings such as fear, anger and mistrust that increase or decrease based on the user's inputs (Colby et al, 1971).

The first leap in chatbot technology came with the introduction of the Artificial Linguistic Internet Computer Entity (ALICE) in 1995, winning the title of "most human computer" at the annual Turing Test contests in 2000, 2001 and 2004 (Wallace, 2009). ALICE used a simple pattern-matching algorithm with its whole intelligence specified via the Artificial Intelligence Markup Language (AIML), an XML-based markup language which allows developers to alter the facts that are known to ALICE (Wallace, 2009; Marietto, 2013).

After that, personal assistants such as Apple Siri, Amazon Alexa and Google Assistant were conceived (Adamopoulou & Moussiades, 2020). Personal assistants refer to systems that can understand user requests and autonomously respond to them by performing actions in the real world (Santos et al, 2018; Silva, 2020). These systems also count as chatbots as conversation is key in the human-to-computer interaction (Luger & Sellen, 2016; Silva, 2020). However, these conversational agents failed to meet user expectations as users were unable to ascertain the system's level of intelligence, are reluctant to use these for complex activities and needing to keep watch of these agents on all but the simplest tasks (Luger & Sellen, 2016).

In 2023, ChatGPT was released for public use via a web user interface (UI) (ChatGPT, 2024). As of writing, ChatGPT uses the GPT-3.5 large language model for its free plan and GPT-4 for its paid plan (ChatGPT, 2024). Research has shown that the question answering performance and general reliability of the Generative Pre-trained Transformer (GPT) model had been continuously improving over time (Tan et al, 2023). Other research has shown how this chatbot can be successfully applied to various domains such as identifying cybersecurity threats, assisting the development of new natural language processing (NLP) models, and providing customized feedback to students and medical professionals (Kalla et al, 2023).

In the same year after the release of GPT-4, an independent team of researchers had set off to make an open-source Large Language Model (LLM) named GPT4All, but later pivoted to lowering the barrier of entry for ordinary people to access the LLMs released by the open-source community (Anand et al, 2023). Their contributions for the latter include APIs that provide high-level yet stable APIs for downloading and using the highest-performing open-source LLMs, compressing those LLMs to be usable in consumer-grade hardware as well as creating a simple GUI application for interacting with the models as chatbots without requiring any code to be written. As of writing, this application is downloadable from the GPT4All website and supports opening documents within the program to enable the chatbot to answer questions using the information contained in the documents (Nomic AI, 2024).

In summary, CQAs are a non-chatbot QAS solution that accepts questions expressed in natural language but suffers from major downsides. Chatbots that have been developed over the decades include ELIZA, ALICE and ChatGPT, and these can be used as QASes. GPT4All provides tools that simplify the process of using LLMs as chatbots.

1.2.4 Question Answering System Language Models

Improvements to chatbot technology were catalyzed by key improvements in language models. This subsection will review papers about language models ranging from statistical language models to large language models used in chatbots today. A summary of the four language model (LM) generations is shown in Figure 6. Zhao et al (2023) note that the years in the figure are determined by the publish date of the most representative studies at each stage, thus the time periods may not be fully accurate. Additionally, due to spatial limitations, not all representative studies are listed in the figure.

The very first type of language model is the statistical language model (SLM), developed in the 1990s (Gao & Lin, 2004). Statistical language models simply predict the next word based on the few words that came before it. These were good enough to enhance IR tasks and general NLP tasks (Liu & Croft, 2005; Zhai, 2008), but these models suffered from the curse of dimensionality as predicting the next word based on previous words required exponentially more computational power and memory (Zhao et al, 2023). There exists some research that tried to alleviate the data sparsity problem of SLMs by introducing techniques such as back-off estimation and Good-Turing estimation (Zhao et al, 2023).

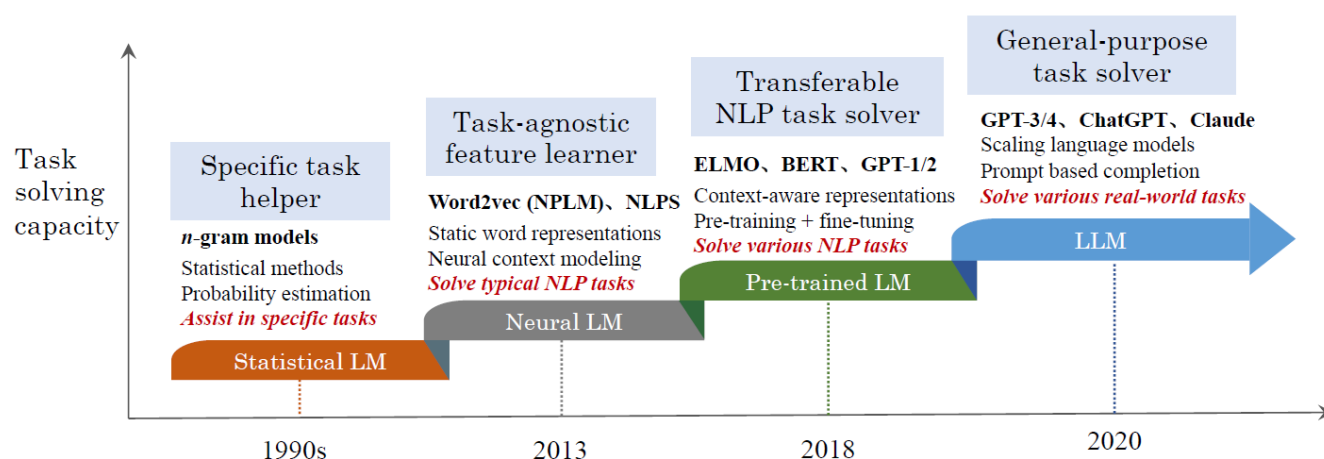


Figure 6. Summary of the evolution process of four language model (LM) generations from the perspective of task solving capacity. Source: (Zhao et al, 2023).

Around a decade later, neural language models (NLMs) were introduced, which were language models powered with neural network technology (Bengio et al, 2000). A major contribution in this research was the concept of distributed representations of words, and word predictions being conditioned on aggregated context features (Bengio et al, 2000; Zhao et al, 2023). Word2Vec was also proposed, aiming to create neural networks that learn distributed word representations, which were shown to be highly effective across a wide range of NLP tasks (Mikolov et al, 2013; Zhao et al, 2023).

There are two major neural network architectures in neural language modelling, which are feed-forward networks and recurrent networks (Goldberg, 2016). For feed-forward networks, the innovation made in this domain was the emergence of Convolutional Neural Networks (CNNs), which contain convolution and pooling layers that help in treating the same sequences of words in a document as the same regardless of their absolute positions (Goldberg, 2016). CNNs have been shown to achieve comparable, if not better results in the field of question answering compared to baseline systems (Dong et al, 2015). CNNs allow encoding arbitrarily sized items into fixed-size vectors that can capture salient features, but most of the structural information in the text is lost, as well as the relations between words that are far apart (Goldberg, 2016).

Recurrent Neural Networks (RNNs), on the other hand, allow this information to be preserved by allowing the model to return another set of output vectors that are then fed back into the model as additional input along with the next token (Goldberg, 2016). RNNs have shown strong results for language modelling, bringing significant improvements compared to SLMs (Auli & Gao, 2014; Goldberg, 2016). However, these networks suffer from major problems including longer training times and limited effective working memory due to vanishing gradients (de Mulder et al, 2015; Goldberg, 2016; Zhao et al, 2023). Some solutions that are used for tackling this issue include batch-normalization, stepwise training, cutting down the size of RNNs or using a specialized architecture such as Long Short-Term Memory (LSTM) (Goldberg, 2016).

In 2018, the first pre-trained language model (PLM), Embeddings from Language Models (ELMo) was proposed (Joshi et al, 2018). Unlike NLMs, pre-trained language models are task-agnostic – they can be adapted to work with a wide range of tasks with the pre-training and fine-tuning paradigm (Minaee et al, 2024). ELMo uses RNN-like encoders as text embeddings, but instead of learning fixed word representations, ELMo models are heavily pre-trained to capture context-aware word representations, then shared with other researchers to be fine-tuned later to suit specific downstream tasks via transfer learning (Zhao et al, 2023). However, this model is still unable to capture long-distance context as the hidden states of the model have limited memory (Zhao et al, 2023). Research had been done to improve this model's performance by adding modules for fluency repair and context matching, but this increases the resources required by the model (Jin et al, 2024).

Shortly after ELMo's introduction, the Bidirectional Encoder Representation from Transformers (BERT) was introduced, demonstrating superior performance compared to ELMo via the use of Transformer encoders (Devlin, et al, 2018; Zhao et al, 2023). Transformer encodings are a radically different technique compared to RNNs and

CNNs, using attention mechanisms that can model dependencies between inputs and outputs regardless of distance, and enabling computational parallelization to reduce training time (Vaswani et al, 2017).

Around the same time that BERT was created, the Generative Pre-trained Transformer (GPT) was also conceived (Radford et al, 2018). GPT is a multi-layer Transformer decoder model pre-trained over an exceptionally large corpus of text, then continuously tries to predict the next word given the preceding words of the context (Jin et al, 2024). Decoder-only models have been shown to demonstrate superior performance after being pre-trained across various tasks (Jin et al, 2024).

As BERT is an open-sourced PLM, it has been derived into different variants over the years, with notable variants including the Robustly Optimized BERT Pretraining Approach (RoBERTa) (Liu et al, 2019) as well as Bidirectional and Auto-Regressive Transformers (BART) (Lewis et al, 2019), both of which have been open-sourced. BART has been shown to outperform RoBERTa in question answering tasks (Lewis et al, 2019).

A Lite BERT (ALBERT) was also open sourced in 2020 and distributed in assorted sizes. ALBERT-xxlarge, the largest ALBERT model, can achieve state-of-the-art performance while having fewer parameters than BERT-large, but at the cost of more computing power required (Lan et al, 2019). Smaller ALBERT models aim to alleviate problems related to GPU and TPU memory limitations as well as long training times, as they take less steps to train than the equivalent size of BERT models – about 15% less for ALBERT-base and 40% less for ALBERT-large (Lan et al, 2019).

The last notable BERT variant that will be reviewed in this paper is DistilBERT, which was open sourced in 2019 (Sanh et al, 2019). Using knowledge distillation, DistilBERT is 40% smaller and 60% faster than the equivalent BERT model, while retaining 97% of BERT's understanding capabilities, which can be a compelling option for edge applications (Sanh et al, 2019).

Research has later shown that scaling up Pre-trained Language Models (PLMs) often leads to improved model capacity on downstream tasks (Kaplan et al, 2020; Zhao et al, 2023). This led to the creation of GPT-3 that takes in 175 billion parameters, compared to the 330-million-parameter BERT and 1.5-billion-parameter GPT-2 models (Brown et al, 2020). GPT-3 has been shown to be capable of solving few-shot tasks through in-context learning which its predecessor GPT-2 cannot do well, especially tasks that require reasoning or domain adaptation (Brown et al, 2020; Zhao et al, 2023). These massive PLMs have been referred to as Large Language Models (LLMs) (Shanahan, 2024), which have attracted significant research attention due to their emergent behaviors (Zhao et al, 2023). Some of the abilities gained by these models include in-context learning, instruction following and multi-step reasoning (Minaee et al, 2024). Zhao et al (2023) note however that there currently lacks a deep, detailed investigation into why LLMs gain abilities that cannot be achieved with typical PLMs.

GPT-3.5 was released soon after by OpenAI in July 2021 (Zhao et al, 2023). This model was fine-tuned on a large corpus of GitHub code in addition to the corpus GPT-3 was trained with, leading to significant performance improvements to solving code and math problems (Drori, 2022; Zhao et al, 2023).

GPT-4 on the other hand was released in March 2023 (OpenAI, 2023), showing a significant improvement on many tasks and outperforming ChatGPT's GPT-3.5 LLM on a diverse range of difficult tasks (Bubeck et al, 2023). GPT-4 was noted to have unparalleled mastery of natural language in tasks such as summarizing texts and question answering in various domains ranging from programming to law, accounting, medicine, music, and other fields (Bubeck et al, 2023). Despite its human-like performance, it too has limitations such as the lack of an "inner dialogue" in problem-solving, low working memory for math problems and performing poorly in constrained writing where the constraints apply globally for the entire generated text (Bubeck et al, 2023).

In 2023, Meta AI released their set of LLMs called LLaMA to the research community (Touvron et al, 2023). These models were created in response to a study done by Hoffmann et al (2022) who have shown that smaller models trained on more data can be competitive with the largest LLM models while also using less computing power. LLaMA pre-training is also done using only publicly available datasets, which allows them to open-source their models and distribute model weights under a non-commercial license (Touvron et al, 2023; Minaee et al, 2024). As such, many derivatives of LLaMA also exist, including Mistral-7B which outperforms LLaMA-2-13B while maintaining efficient inference (Jiang et al, 2023; Minaee et al, 2024). The first iteration of the GPT4All model was also a fine-tuned LLaMA-7B model, which was found to yield less perplexities on ground truths compared to Alpaca-Lora (Anand et al, 2023).

PaLM was also published that same year, the first large-scale model that utilizes both the Transformer and Pathways technologies to achieve efficient scaling during training (Chowdhery et al, 2023). This LLM is also able to demonstrate chain-of-thought reasoning, including explaining jokes and complex questions (Chowdhery et al, 2023). Just like other LLMs, PaLM also has been derived into many variants as shown in Figure 7. One notable variant is Med-PaLM, a domain-specific PaLM designed to provide high medical question answering performance (Minaee et al, 2024). Med-PaLM itself was improved into Med-PaLM 2 via fine-tuning and ensemble prompting, setting a new state-of-the-art on the MedQA dataset with a score of 86.5% (Minaee et al, 2024).

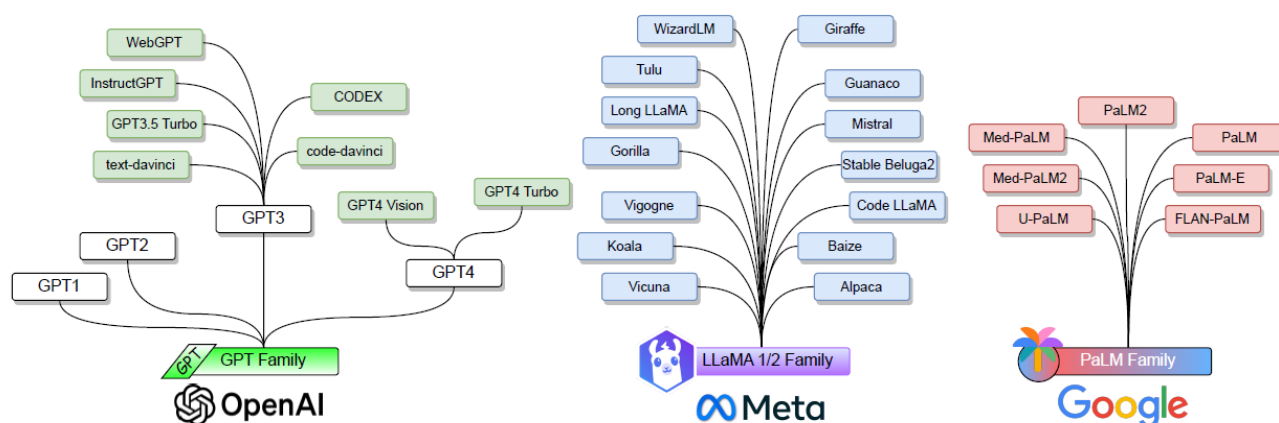


Figure 7. Popular LLM Families. Source: (Minaee et al, 2024).

In summary, SLMs were the first category of language models that were developed, followed by NLMs, PLMs and finally LLMs. New techniques such as Word2Vec and Transformers also significantly contributed to the performance of these language models, while the spirit of open source contributed to the performance of future language models.

1.2.5 Techniques of Using Language Models with Data

This section will discuss how language models can be customized with data for purposes like answering questions over a specific dataset. There are four distinct ways to feed data to a language model, which in ascending order of computational power required are prompt engineering, retrieval augmented generation, fine-tuning and lastly pre-training.

Prompt engineering refers to the technique of programming an LLM using only prompts to perform a specific behavior (White et al, 2023). Prompt engineering is possible because of the capabilities of today's LLMs to follow instructions and rules specified in the prompts sent to them, with sufficiently engineered prompts being capable of creating novel paradigms of interactivity, such as simulating a Linux terminal (White et al, 2023).

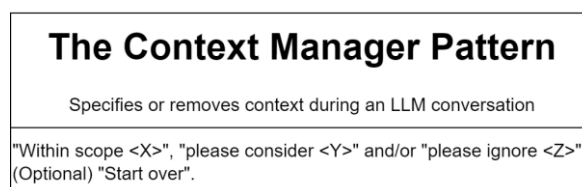


Figure 8. White et al's Context Manager Pattern. Source: (White et al, 2023).

Using this method, it is possible to create a QAS for the Sarawak Gazette using White et al's (2023) Context Manager Pattern as described in Figure 8, specifying that the following questions sent to the LLM should be answered with the context of Sarawak Gazette. As no other data input is fed into the LLM, all the generated answers will be solely based on the model's parametric memory. Because of this, the questions that are answerable to the LLM will be limited to what the model has seen during training – public data that was current and available at the time of training (Tebbe, 2024; Q&A with RAG, 2024).

Retrieval Augmented Generation (RAG) is another technique for making a language model perform question answering. Compared to prompt engineering, RAG involves external data being supplied to the LLM as context for the LLM to refer to, rather than having the LLM rely solely on the data it was fed during training.

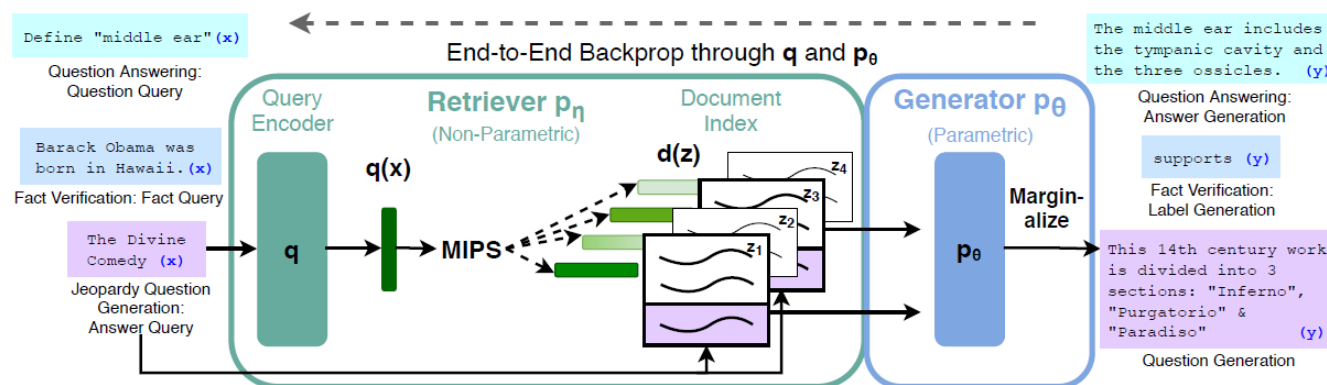


Figure 9. An example RAG architecture. The study used BERT-like PLMs as the language models for the architecture instead of LLMs, which allows fine-tuning to be done in addition to RAG for improving the accuracy of the RAG system without being prohibitively expensive. Source: (Lewis et al, 2020).

An example of an architecture that utilizes RAG is shown in Figure 9. The retriever component here is composed of two sub-components, which are the query encoder and the document indexer. The query encoder (labelled q in Figure 9) takes the input query and encodes it into an embedding vector that captures the essence of the query, which in the study is done with BERT. On the other hand, the document indexer sub-component retrieves the documents which have embeddings (labelled $d(z)$ in Figure 9) that are the most similar to the embedding of the input query. Any metric can be used to compute the similarity – in Figure 9, Maximum Inner Product Search (MIPS) is used to determine the top-ranking documents. Lastly, the generator component takes in the retrieved documents and query, then uses them to generate a novel output.

In this architecture, two language models are involved – one for the retriever and another for the generator. The documents are part of non-parametric memory, meaning that they are not memorized inside any of the language models' parameters, instead being an external source of knowledge. This in turn allows the RAG system's knowledge base to be updated simply by replacing the input documents supplied without needing to perform additional model fine-tuning (Lewis et al, 2020).

Fine-tuning is a technique where LLMs (or any PLMs in general) that have acquired the general representations of language are retrained over new data to be adapted to a specific downstream task (Devlin et al, 2018). Compared to training a language model architecture from randomly initialized parameters, fine-tuning is less computationally expensive while still being able to achieve state-of-the-art performance (Devlin et al, 2018).

Figure 10 shows the pre-training and fine-tuning steps done for BERT for question answering. While pre-training must involve a huge data set, which in this case involved over 3 billion words from BooksCorpus and the English Wikipedia (Devlin et al, 2018), fine-tuning can be done with far smaller datasets involving only thousands of examples (*Retrieval Augmented Generation*, 2024). Nevertheless, larger datasets will give more stability, for example datasets with hundreds of thousands of examples are far more tolerant of what hyperparameters are used for fine-tuning (Devlin et al, 2018). The same study's authors mention that the right hyperparameter values to use when fine-tuning BERT are different for each task, but notes that a batch size of between 16 to 32, learning rate between $2e-5$ to $5e-5$ and fine-tuning on the new dataset for 2 to 4 epochs will generally work well across all tasks.

The most powerful and most expensive technique for creating a question answering system using an LLM is to train it from the ground up, with randomly initialized model weights over an exceptionally large corpus. Not only will this require deep expertise on big data gathering and processing, but there are also engineering challenges involved in the parallelization and distribution of model training for developing a capable LLM, which are difficult enough such that researchers must either be engineers or be working with engineers to solve such problems (Zhao et al, 2023). For example, pre-training the LLaMA-7B model on A100-80GB GPUs required over 1 trillion input tokens and over 80 thousand GPU-hours (Touvron et al, 2023). However, despite the complexity of pre-training, this method

would provide the maximum amount of control over the model's performance as all weights are fully controlled by only the researcher's team, in comparison with fine-tuning where the general representations of language have been pre-trained by others (Devlin et al, 2018).

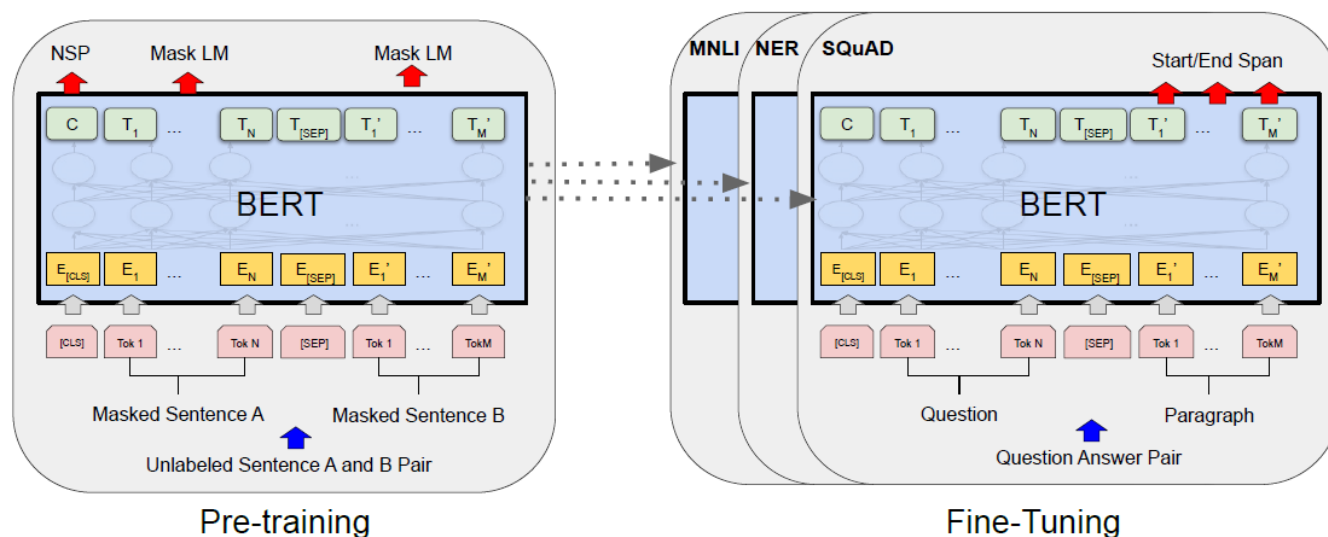


Figure 10. BERT pre-training and fine-tuning. Fine-tuning the model utilizes the same architecture as pre-training, only differing in output layers and using the pre-trained model's weights rather than randomly initiated weights. Source: (Devlin et al, 2018).

In summary, the four distinct ways to customize LLMs to data are prompt engineering, retrieval augmented generation, fine-tuning, and pre-training. Prompt engineering requires the least computational power but also offers the least control, while pre-training offers the most control but demands the most computational power.

1.2.6 Evaluation of Question Answering Systems

This section will first discuss the measurement categories used for determining the performance of QASes, then present other studies' findings regarding the methods used for evaluating these systems.

One study argued that for QASes to be useful, developers should set performance goals based on the distribution of correct answers over ranks rather than measuring a single mean reciprocal rank value, as it would lead to increased user satisfaction (Kokubu et al, 2005; Ong et al, 2009). Based on this, User Satisfaction with Question Answering Systems (USQAS) was proposed as a user-centered framework for evaluating QASes, composed of four main elements which are Usefulness, Ease of Use, Information Quality and Service Quality (Ong et al, 2009).

A 2020 study notes that all chatbot evaluation metrics used by prior studies fall under three broad categories, which are satisfaction, efficiency, and effectiveness (Casas et al, 2020). That study's authors state that for studies where algorithmic performance is critical to the application, such as question answering, those studies often neglect evaluating efficiency and satisfaction as they are irrelevant to the problems they are seeking to solve, instead solely focusing on effectiveness, which is the correctness of the system. They also establish that no metrics have been used more frequently than others for evaluating chatbots, likely because there is currently no golden standard for evaluating chatbot performance. In fact, there are many papers that only present their chatbots and neglect rigorously testing them (Casas et al, 2020).

For the case of Thapliyal's (2023) thesis that involved building a QAS for historical books, there were a wide variety of metrics used including precision, recall and cosine similarity. In the context of evaluating the relatedness between each reference answer and generated answer, precision measures the proportion of the generated answer covered by the reference answer, while recall measures the proportion of the reference answer covered by the generated answer (Thapliyal, 2023). These two metrics are not the same – precision is reduced when the generated answer has words or tokens that are not present in the reference answer, while recall is reduced when the generated answer does not have words or tokens that are present in the reference answer. On the other hand, cosine similarity measures the two answers by their embedding vectors rather than tokens, with high cosine similarity indicating that the generated answer is relevant (Thapliyal, 2023). Each one of these evaluation metrics cover a distinct perspective:

precision checks for extra predicted tokens, while recall checks for missing predicted tokens and cosine similarity checks for the difference of meaning between the reference and predicted sentences.

A 2016 study examined the usefulness of other automatic evaluation metrics such as Recall-Oriented Understudy for Gisting Evaluation (ROUGE), Bilingual Evaluation Understudy (BLEU) and METEOR for evaluating the performance of chatbots (Liu et al, 2016). Their findings show that for unsupervised dialogue systems where multiple responses are valid, these metrics correlate very weakly against scores assigned by humans. The study's authors note that an alternative, automatic evaluation metric that is representative of human-assigned scores remains an open research question. A 2022 study that applied chatbot technology to make a historical industrial heritage QAS cited the former study as the reason they did not use BLEU for evaluating their chatbot (Arcan et al, 2022).

More recent evidence (Deutsch et al, 2021) highlights that ROUGE does not truly measure the amount of information that is common between two pieces of text, instead this metric simply measures the degree of overlap between them. Pieces of information that are completely different can still result in a high ROUGE score if they happen to contain the same words or tokens, which is undesirable behavior for an evaluation metric.

Other than these automatic evaluation metrics, it is also possible to use PLMs and LLMs to evaluate QASes by prompting them to give a rating based on how good an answer is for a given question. An example of this is the Vicuna evaluation pipeline (Zheng et al, 2023). However, there are some well-known criticisms regarding the usage of language models (LMs) to evaluate the generated outputs of another language model. Liu, Moosavi and Lin (2023) have shown that evaluator LMs tend to assign higher scores to generator LMs of the same family. An example of this phenomenon is BARTScore giving higher scores to BART-generated text compared to other LMs. In addition, another study (Wang et al, 2024) has shown that LLMs exhibit positional bias, where merely shuffling the order of content to be evaluated can result in different evaluation scores.

In summary, there is currently no golden standard for evaluating the performance of chatbots, with different researchers using different methods of evaluation depending on the specific problems their studies are addressing. The only evaluation metrics that are usable in this study are those that have been used by prior studies about applying chatbot technology to historical archives, as using other evaluation methods runs the risk of this study's results being invalid or cannot be compared meaningfully with prior studies.

1.2.7 Use of Chatbots over Historical Texts

Very few studies could be found that involve both chatbot technology and using it over historical documents. All four papers found that are related to both chatbots and historical text will be reviewed in this section, in chronological order.

A 2005 study investigated the application of chatbot technology in animating corpora (Shawar & Atwell, 2005). Specifically, they used SGML annotated data from the British National Corpus (BNC) and converted them into AIML for training the ALICE chatbot, along with parsing user questions using a most significant word approach to increase the possibility of finding an answer. They also investigated using a Frequently Asked Questions document of University of Leeds' School of Computing for training the chatbot in the same way.

The results of the study were that ALICE's pattern-matching approach did allow ungrammatical or ill-formed inputs to still yield the correct answer, but the chatbot's replies were sometimes inconsistent, ungrammatical, or otherwise odd due to the variety of speakers in the corpus. Nevertheless, they note that ALICE can at least qualitatively visualize the use of language in the training corpus, such as gender variation, domain-specific terminology, and the sentiment of certain topics.

Another study in 2013 looked at designing a chatbot for emulating a historical figure (Haller & Rebedea, 2013). To use this chatbot, the user must first select their desired historical figure, or key in the Wikipedia and DBpedia links of a historical figure for the chatbot to internally generate a list of facts relating to the figure. It is after this step that the user can ask questions related to family relations, political views, military career, and a few others. The chatbot, which is built with ChatScript, then tries to pattern-match the user's question to determine what the question is asking. If there is a match, the chatbot gives an answer that is adapted based on the facts stored in the chatbot's memory. The authors note that since the chatbot is meant to be used by the public, a user-friendly interface is an absolute necessity.

A noted problem with this chatbot is that if the user's question fails to be pattern-matched, the chatbot returns a random statement from the knowledge base. Also, all ChatScript rules that are used by the chatbot must be typed by hand, which means that historical figures who were inventors, philosophers, doctors, and other careers require additional hand-typed rules to deal with the additional questions that could be asked.

A more recent study conducted in 2022 demonstrated how a chatbot can be built in the domain of 18th to 19th century industrial heritage (Arcan et al, 2022). The authors algorithmically generated the questions by extracting the most meaningful terms from articles in Project Gutenberg and the British Digital Library Collection using Saffron, extracting the relation of those terms using Stanza, sent each relation as a query to the libraries to obtain relevant training data, then used OpenNMT to convert the extracted terms into questions. The chatbot also utilizes a custom-made Transformer neural language model (NLM) architecture for answer generation. They did not use a pre-trained language model (PLM), instead opting to build a language model from scratch.

The resulting chatbot was manually evaluated with four volunteers. In general, the chatbot's answers ranged between acceptable and possibly acceptable, sometimes giving only half-answers or incomprehensible text. The limitations noted by the authors include the term and relation extraction method, data collection method and evaluation method, as they did not include relevant historical expertise in their study. Their methods also only extracted singular terms for question generation, and the questions themselves are simply generated from a template. They note that the usage of Transformer models such as Text-To-Text Transfer Transformer (T5) may be more effective for question generation.

The latest research found by this paper's authors relating to applying chatbot technology to historical text is a 2023 thesis that applies a series of AI-powered question answering models for the Mahabharat (Thapliyal, 2023). For this paper, emphasis will be given to that author's Retrieval Augmented Answer Generation System (RAAGS), which is similar to a QAS except that context does not need to be provided by the user, as the model has been pre-trained with the context. This means the user only needs to input their question to receive an answer, which very closely resembles the question answering capabilities of chatbots on usual questions.

Using T5 for the RAAGS, that author achieved a cosine similarity of 0.763. Cosine similarity was introduced as a metric that can be used in evaluating the quality of generated answers by comparing the vector embeddings of the reference answer and generated answer for each question. The RAAGS system is inferior compared to the 0.827 cosine similarity achieved by that author's contextual QAS, which is expected due to the lack of user-fed context. The same author also claims that the RAAGS can be extended to other languages and historical books to make historical texts more accessible (Thapliyal, 2023). Future directions suggested by that author include looking into issues of domain-specificity, how to evaluate the systems more effectively, how to increase the cost-effectiveness of the systems and using bigger LLMs for building the systems.

In summary, a variety of approaches have been used for building chatbots on historical documents, including pattern-matching methods, a custom-built NLM and the T5 PLM. Some issues with early chatbots include inconsistencies and patterns needing to be manually specified, while later chatbots have issues related to evaluation, training dataset generation and integrating bigger models into their systems.

1.2.8 Discussion

Limited research has been done on the Sarawak Gazette, with the latest research only focusing on collecting data required for the semantic web (Rosita et al, 2010; Ramli et al, 2017), which would allow for the building of IR systems to retrieve documents from the Sarawak Gazette. But the literature has shown that information retrieval is different from question answering, as QASes must return answers to questions that are straight to the point rather than the relevant documents, saving the user's time (Allam & Haggag, 2012; Ojokoh & Adebisi, 2019). As such, this research which applies chatbot technology to create a QAS with the context of Sarawak Gazette is a novel research path that may improve the accessibility of information contained in the Sarawak Gazette.

While pre-training an LLM from scratch would theoretically yield the best accuracy, this requires a corpus far too huge given the resource constraints of this study. Fine-tuning also cannot be done as this study's Sarawak Gazette dataset has too few examples, as well as fine-tuning itself requiring too much computational power (Ling et al, 2023). Therefore, the best path forward for this research to make a QAS that leverages chatbot technology is Retrieval

Augmented Generation. Considering this, the architecture of this study will follow the RAG architecture introduced by Lewis et al (2020), with only a few parts modified to match the objectives of this study.

As there is currently no gold standard for chatbot evaluation (Casas et al, 2020), evaluation will be done using precision, recall, accuracy, F1-score, cosine similarity and BLEU-1 to cover the various aspects of the QAS while also being meaningfully comparable with a previous study on historical document question answering.

2 METHODOLOGY

This section details the research strategy of this study. In general, the steps involved with building a QAS included data preprocessing, implementing the reader/retriever architecture and displaying a user interface (UI), whereas the steps involved with creating the reference Sarawak Gazette QAS included dataset generation and validation.

A Large Language Model (LLM) was used twice in the methodology. Its first use was to generate the questions and reference answers as described in Section 2.2, which afterwards were checked by humans to ensure they are correct. Then, these questions were fed into the QAS described in Section 2.1 to get generated answers, which afterwards were evaluated against the reference answers using automated metrics instead of humans, such as accuracy and BLEU.

2.1 The Question Answering System for the Sarawak Gazette

This study closely followed the research steps made by Thapliyal (2023) due to its similarity. In general, building the QAS involved the processes of dataset acquisition, data preprocessing, context retrieval, retrieval augmented generation, model evaluation and finally UI construction. The ultimate goal of this study was to make a question answering chatbot focused on the Sarawak Gazette using the LangChain library.

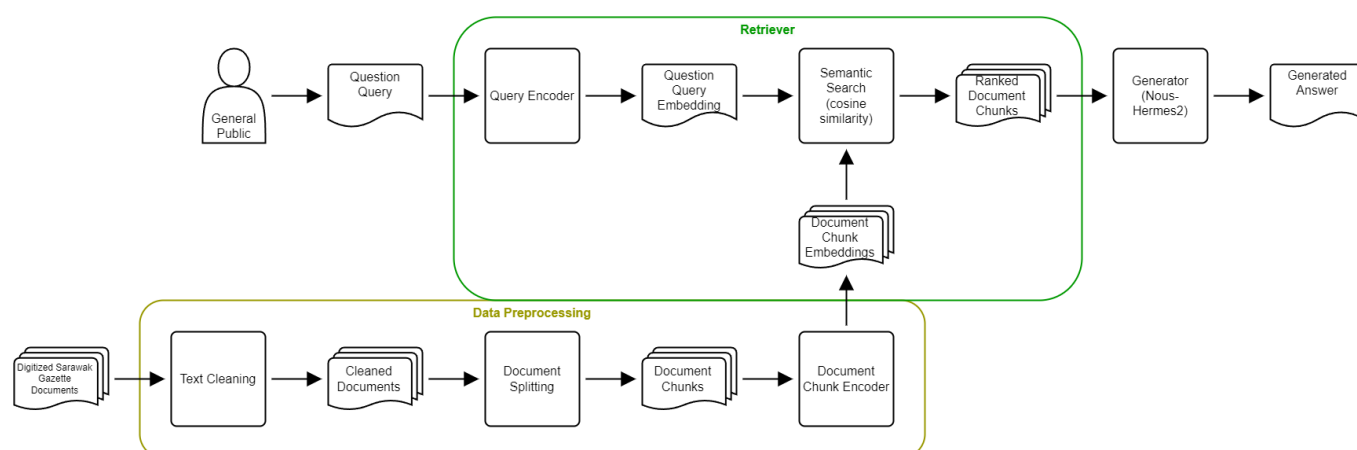


Figure 11. Overall architecture of this study's Question Answering System.

Figure 11 is a summary of this study's methodology. The architecture used is an adaptation of the RAG architecture introduced by Lewis et al (2020), with the main difference being the addition of data preprocessing steps to improve the quality of the input documents.

The very first step was to obtain the digitized Sarawak Gazette documents in raw text format as seen at the bottom-left of Figure 11. The process of obtaining this data falls under data acquisition. Due to resource constraints, only a subset of all Sarawak Gazette papers was used for this study, specifically papers that had been digitized and were readily accessible by the study's authors. The composition of the files used in this study by publish year is shown in Table 1. It should be noted that the table does not show the number of distinct tokens – tokens that appeared more than once were counted for each instance.

There did not appear to be a hard limit on either the minimum or maximum amount of digitized text documents that needed to be collected. However, if a specific task involves a dataset that the language model has not seen, the model will not be able to perform that task (Thapliyal, 2023). Extrapolating from this, if the input documents did not contain the information required to answer certain domain-specific questions, then the resultant chatbot would also not be able to answer those questions.

Table 1. Statistics of digitized Sarawak Gazette text files used in study.

Year	Issues Covered	Total Tokens
1907	11	248,237
1919	22	226,873
1920	27	256,913
1922	12	252,174
1926	12	280,533
1927	11	214,923
1928	12	121,251
1929	6	93,030
1930	12	271,482
1931	11	246,402
1932	12	189,186
1933	12	112,095
1934	11	65,673
1935	12	187,264
1936	12	266,664
1937	12	210,850
1939	1	13,978
1941	1	16,862
1947	12	260,970
1948	12	275,835
1949	1	28,910
1951	1	18,168
1953	10	212,656
1958	4	107,790
Grand Total	249	4,178,719

The next step involved data preprocessing, which in this case involved data cleaning, dataset splitting, feature extraction and creating a vector store. Data cleaning involved removing special characters and excess space characters that were present due to imperfections in digitizing the Sarawak Gazette. Other steps that would normally be done for data cleaning such as tokenization and part-of-speech tagging were unnecessary as the generator automatically performed these steps when given textual input. Case folding was also not done because the chatbot needed to know the capitalization of words to properly format its output.

Thapliyal's (2023) study used LangChain to split up the text documents, creating chunks with an average size of 8000 characters per chunk with 300 characters of overlap, which yielded an average of 699 words per chunk. However, the author also states that choosing the right context chunk size is difficult and requires experimentation. While more context would increase the accuracy of the generated answers, not only will longer prompts incur more processing time, but LLMs forget tokens that exceed their context window, for example Mixtral 8x7B has a context length of 32 thousand tokens (Mistral AI Team, 2024).

This study used the same library, LangChain, to split the text data into smaller chunks. The chunks overlapped by 300 characters just like in Thapliyal's (2023) thesis, but the average number of characters per chunk was reduced to 1200, as it was found that any higher numbers than this would always cause this study's chosen LLM, Nous-Hermes2, to halt due to too many input tokens. This is because the GGUF file format, which is the file format used by GPT4All models, imposes a context length limit of 2048 tokens (GPT4All, 2024).

Feature extraction of the document chunks was required for determining which smaller chunks were the most relevant for answering the user's question. This is done via the process of vectorization, where the context chunks are embedded into a very high-dimensional vector space such that texts that are semantically close together are also close together in the vector space (*Embeddings*, 2024). This allows measurements to be made between two pieces of text by comparing their similarity in the vector space, allowing operations such as semantic search to return the n -th most similar pieces of text from a given query (*Embeddings*, 2024; *Text embedding models*, 2024).

This study again used the LangChain Python library to generate vector embeddings from context chunks and cached the generated embeddings into a vector store, referred to as the Embedding-Chunk Database in Figure 11. The latter process was so that the embeddings did not need to be regenerated whenever a different user question was asked.

After the data preprocessing phase, efforts were made at implementing the Context Retrieval component. This component was an IR system that served to take in a user's question, then give a ranked list of document chunks that were the most relevant to the user's question, hopefully capturing enough context for the QAS to then generate the correct answer using the returned chunks. This was done by generating the vector embedding of the user's question, then comparing the question's embedding with all the chunk embeddings in the vector store. Embeddings that had the highest similarity with the question were put into a ranked list and returned by the IR system.

The parameters available for this component were the similarity metric to use and the number of chunks to return. For this study, only the top 5 most similar chunks were returned, and similarity was measured as the cosine similarity between the question and chunk embedding vectors, just like in Thapliyal's (2023) study.

After that, the ranked document chunks were used to generate an answer to the user's question using the Retrieval Augmented Generation (RAG) technique. To do this, the ranked retrieved contexts and the user's question were combined into a chatbot prompt, as shown in Figure 11. The structure of the prompt itself used LangChain's most commonly used structure for retrieval augmented question answering ("*rlm/rag-prompt*"), which asks the user's question, gives the context behind the question and has additional sentences telling the chatbot to say that it doesn't know when the contexts cannot answer the question, as well as telling the chatbot to keep the answer concise (*rlm*, 2024).

LangChain's GPT4All module was used for answer generation, as it allows an LLM to be run locally on consumer-grade hardware and perform the following actions for each user question:

- Send the constructed prompt into the LLM,
- Retrieve the output of the LLM, and
- Send the response back to the program.

This study's program then handled the response by displaying it to the user, completing the application flow. The GPT4All API itself supports a wide variety of LLMs, not just GPT (Anand et al, 2023). While OpenAI does offer an API to send prompts and receive answers from GPT-3.5, their model is not open source and is only licensed for commercial use, requiring payment (*Pricing*, 2024). The authors of this study have chosen to use the Nous-Hermes2 LLM since as of writing, it is a state-of-the-art chatbot model – on the PIQA dataset, it outperforms LLaMa-13B by 1.6 points and GPT-J-6.7B by 4.5 points (Anand et al, 2023). This is also true on the OBQA dataset, outperforming LLaMa-13B by 4.0 points and GPT-J-6.7B by 8.2 points (Anand et al, 2023). The usage of an LLM makes this study distinct from Thapliyal's (2023) study, as the author of that study used the T5 PLM, which does not leverage chatbot technology for answering the user's question.

Evaluating this study's chatbot involved the testing dataset generated from this study, which is described later in this paper. Each question from this testing dataset was fed into the question answering system to get a corresponding generated answer for the question. The authors of this study used average precision, average recall, average accuracy, average F1-score, BLEU-1 and average cosine similarity between question and answer to quantify chatbot performance, as many of these metrics were also employed by a previous study (Thapliyal, 2023). As mentioned in Section 3, precision measures the proportion of the generated answer covered by the reference answer, while recall measures the proportion of the reference answer covered by the generated answer (Thapliyal, 2023). The exact formula to calculate precision is:

$$precision = \frac{n(S_R \cap S_G)}{n(S_G)} \quad (1)$$

In the formula above, S_R is the set of all tokens in the reference answer, S_G is the set of all tokens in the generated answer and n is a function that gives the cardinality of the set.

Similarly, the exact formula to calculate recall is:

$$recall = \frac{n(S_R \cap S_G)}{n(S_R)} \quad (2)$$

S_R , S_G , and n in the formula above have the same meaning as in the formula for calculating precision. Precision and recall will be calculated for each pair of generated answer and reference answer, then both are averaged to get the average precision and average recall, respectively.

Average accuracy was calculated in a similar way, using the following formula:

$$accuracy = \frac{n(S_R \cap S_G)}{n(S_R \cup S_G)} \quad (3)$$

For F1-scores, the below formula was used, with precision and recall calculated using equations 1 and 2.

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \quad (4)$$

BLEU-1 is significantly more complicated to calculate. Papineni et al (2002) provides a more complete description on calculating BLEU scores between predicted text and reference texts, but in brief:

$$BLEU = BP \times \exp\left(\sum_{n=1}^N w_n \log p_n\right) \quad (5)$$

Where BP is the brevity penalty, equal to 1 when the number of predicted tokens is at least the number of reference tokens and $\exp(1-\text{ref_length}/\text{predicted_length})$ otherwise, \exp is the exponential function ($\exp(x) = e^x$), N is the maximum n-gram length, w_n the weightage of the n -th N -gram precision score (usually set to $1/N$) and p_n the precision score of the n -th N -gram. Since only unigrams are considered for BLEU-1, N and thus w_n are substituted with 1 for this study:

$$\begin{aligned} BLEU - 1 &= BP \times \exp\left(\sum_{n=1}^1 1 \log p_n\right) \\ &= BP \times \exp(\log p_1) \end{aligned} \quad (6)$$

Calculating cosine similarity required a different approach. To do this, each question's reference answer and corresponding generated answer were vectorized into embedding vectors, then compared using cosine similarity to score how well the generated and reference answers aligned. A high cosine similarity meant that the reference answer and generated answer were similar, indicating that the generated answer was high quality. Just like Thapliyal's (2023) methodology, the cosine similarities obtained across all generated and reference answers were averaged. This aggregated result was then used to compare this study's chatbot performance with the other study's chatbot.

Another dataset, *ChroniclingAmericaQA* (Piryani et al, 2024), was also used for evaluating the chatbot. This dataset contains testing data for NLP models specifically for historical document question answering tasks, with each example consisting of context, question, and reference answer triplets. The same experimental strategy was used for evaluating the chatbot over these datasets apart from the context retriever, where it was replaced with the context obtained from the triplets instead of being retrieved from the *Sarawak Gazette*. Due to time constraints, only 500 randomly sampled testing examples from the *ChroniclingAmericaQA* dataset were used to evaluate the QAS.

Another dataset apart from the *Sarawak Gazette* was used due to the latter's unique features. An example is that the topics covered in the *Sarawak Gazette* are widely varied, including politics, lifestyle, law, culture, geology, economy, and even flora and fauna (Rosita et al, 2010; Fong & Ranaivo-Malançon, 2015). The *Sarawak Gazette*, while usually written in English, sometimes also contains passages written in historical Malay or Jawi, which can be more

challenging for the chatbot than the Chronicling America newspaper collection, as the latter only consists of historical English text.

For implementing the UI, Qt6 for Python was used. Qt6 is a library for creating UI applications and is licensed under either a commercial license or the Lesser GNU Public License Version 3 (LGPLv3) license. LGPLv3 only requires that the program's code be freely available if the library is statically bundled with the program (*Obligations of the GPL and LGPL*, 2024), which is not the case due to Python's import mechanism. Therefore, the free version of Qt6 was used for building the UI.

2.2 Creating the Reference Question Answering Dataset

As there did not exist a question answering testing database specifically for the Sarawak Gazette, one needed to be created for evaluating the QAS. Dataset creation involved sampling, generation, and validation steps. For this study, the authors had chosen to generate 100 pairs of questions and reference answers, as 100 is the same number used by Arcan et al (2022) for evaluating their own chatbot.

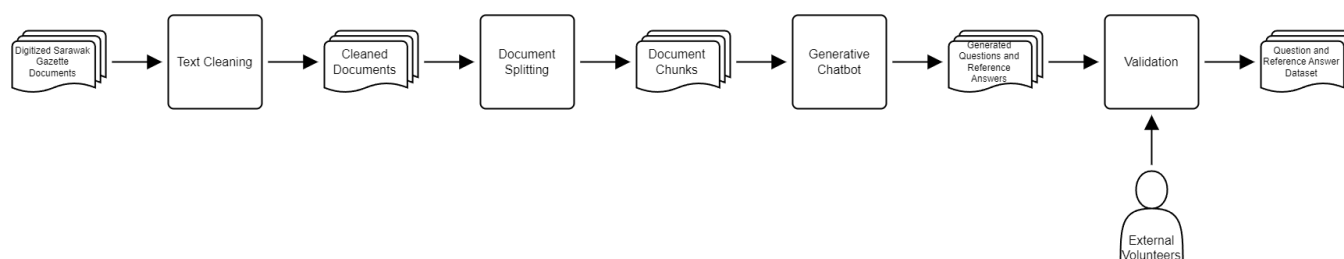


Figure 12. Creating the Reference Question and Answer Dataset.

Sampling was done by taking 100 random passages from the original Sarawak Gazette dataset, each with an average length of 5000 characters. After that, generation was done using a generative chatbot, specifically Nous-Hermes2. Using the entire content of individual gazette papers to generate questions was initially considered, but then deemed to be impossible due to the LLM's token limit. Instead, Thapliyal's (2023) method was used, which was by splitting the documents into smaller chunks, then using these chunks as context to generate the questions and reference answers. Each question involved only one chunk.

The next step was to validate the generated reference dataset via external volunteers who are knowledgeable in historical topics, to ensure that the questions and reference answers are relevant to the Sarawak Gazette and that the answers are correct. All the generated content needed to be verified, as LLMs are vulnerable to generating hallucinated answers, which are answers that appear correct at first glance but are incorrect in reality (Thapliyal, 2023). For this activity, the 100 pairs of questions and reference answers were grouped into 10 sets of 10 questions each, then all sets were evenly distributed across 10 volunteers, one set each.

The quality of the generated questions and reference answer pairs were evaluated with the same methods used by Arcan et al (2022). Specifically, all generated contents were assigned a score from 1 to 3 based on Coughlin's (2003) error classes, which consists of the following:

- 1: Unacceptable. Accurate information is little to none, or the generated content is absolutely incomprehensible.
- 2: Possibly Acceptable. Some information has been generated accurately and the generated content could be comprehended if given enough time and/or context to understand them.
- 3: Acceptable. Not necessarily perfect, but all information is generated accurately and definitely comprehensible.

3 RESULTS

This section presents the figures obtained from this study's experiment, specifically the QAS's performance and the quality of the QAS's testing dataset. The QAS's performance metrics shown in this paper are over two different datasets, which are this study's Sarawak Gazette QA dataset as well as the ChroniclingAmericaQA (Piryan et al,

2024) dataset. On the other hand, the quality of the QAS's reference dataset was determined based on the mean evaluation scores given by 10 annotators.

3.1 Question Answering System Evaluation

The completed QAS for the Sarawak Gazette was tested using two datasets, which are this study's generated Sarawak Gazette QA dataset and the *ChroniclingAmericaQA* dataset. For each example in each dataset, the question was sent to the QAS to get a generated answer. Then, the generated answer and reference answer were used to determine the precision, recall and cosine similarity of the QAS for the given question. In this study, overall performance was determined by taking the mean of these metrics over 100 generated answer and reference answer pairs (Table 2).

Table 2. QAS Performance over Sarawak Gazette Question Answering and *ChroniclingAmericaQA* datasets.

Metric	Sarawak Gazette QA Dataset	<i>ChroniclingAmericaQA</i> Dataset
Mean Precision	0.487	0.076
Mean Recall	0.430	0.645
Mean Accuracy	0.296	0.075
Mean F1-Score	0.436	0.130
Mean Cosine Similarity	0.678	0.311
BLEU-1	0.383	0.070

On the Sarawak Gazette Question Answering Dataset, the QAS had poor mean precision and recall. The low precision was due to the answers generated by the QAS often having tokens that were not present in the reference answer. A possible interpretation of this is that only 48.7% of the words generated by the QAS were relevant. On the other hand, the low recall was due to the generated answers often missing tokens that should be present as per the reference answers, which could be interpreted as the QAS only giving complete and correct answers 43.0% of the time.

When the dataset was switched to *ChroniclingAmericaQA*, mean recall improved while cosine similarity and precision dropped, with mean precision dropping by over 84%. The QAS's generated answers were not concise at all compared to the reference answers in the *ChroniclingAmericaQA* dataset, but they did contain more words from the reference answer on average. The lower cosine similarity on the *ChroniclingAmericaQA* dataset compared to the Sarawak Gazette QA dataset was highly likely a consequence of the exceptionally low precision, because the considerable number of additional words generated by the QAS could have changed the overall meaning of the generated answer to be far from the overall meaning of the reference answer.

3.2 Sarawak Gazette Question Answering Dataset

After the dataset for evaluating the QAS had been generated, 10 annotators with a background of Sarawakian history annotated the dataset. Each annotator rated 10 questions and reference answer pairs by assigning an evaluation score of either 1, 2 or 3 for each item based on Coughlin's (2003) error classes. The mean evaluation scores assigned by each annotator were then averaged to arrive at a final number quantifying the dataset's quality (Table 3).

The generated testing dataset had been consistently scored highly across annotators, with 8 annotators giving a perfect score of 3 for each item. Annotator 6 rated 3 for all items except one, where a rating of 1 was given instead. This yielded a mean evaluation score of 2.8. Annotator 7 only gave 5 items a rating of 3 – one item was scored 1 while the other four were scored 2. This explains the minimum mean evaluation score of 2.4 given by Annotator 7. Even with the presence of two outliers, the higher prevalence of perfect scores had shifted the average mean evaluation score to almost 3.

In summary, this study's QAS performed best on the Sarawak Gazette QA dataset in terms of precision and cosine similarity, while performing best on the *ChroniclingAmericaQA* dataset in terms of recall. However, the maximum evaluation scores attained by the QAS are still quite low – 0.488 on mean precision, 0.646 on mean recall and 0.678

on mean cosine similarity. The Sarawak Gazette QA dataset was rated highly among annotators, attaining an average mean evaluation score of 2.9.

Table 3. Evaluation scores assigned by annotators over generated testing dataset.

Question Numbers Evaluated		Mean Evaluation Score
Annotator 1	1-10	3.0
Annotator 2	11-20	3.0
Annotator 3	21-30	3.0
Annotator 4	31-40	3.0
Annotator 5	41-50	3.0
Annotator 6	51-60	2.8
Annotator 7	61-70	2.4
Annotator 8	71-80	3.0
Annotator 9	81-90	3.0
Annotator 10	91-100	3.0
Average of Mean Evaluation Score		2.9

3.3 Analysis

This section elaborates further about the results obtained in this study, particularly the poor precision and recall of the QAS and the difference in mean cosine similarity over the two testing datasets. This section will also compare this study's results with those of previous studies.

A possible explanation for the poor precision and recall achieved by the QAS on the Sarawak Gazette testing dataset is that there are multiple valid answers for many questions. On the question "What are some of the sections that were featured in the Malaya Borneo Exhibition?", the generated answer was "Stamps, specimens from mines in Selangor and Kinta, large model of the Sarawak Oil Fields at Mid, mineral specimens from the Peninsula, China clay and pottery works at Gopeng, native entertainments such as Hathorn and Borsa performances, Memora presented by Kelantan, a photographic exhibition sponsored by the Sarawak Photographic Society, and an Art Exhibition with categories like drawing, painting in various mediums, wood-carving, metal work, fancy needlework, etc." while the reference answer mentions "Agriculture, Minerals, Amateur Photographic, Professional Photography, Western Arts".

Mines and minerals are different tokens, thus the generated answer both missed a word that was in the reference answer and had a word that was not, even though the QAS did correctly mention a section for minerals. This example yielded precision and recall of 0.290 and 0.500 respectively, while cosine similarity was 0.824.

For the ChroniclingAmericaQA dataset, a possible hypothesis for the lower cosine similarity and extremely low precision is that the LLM always tried to generate complete sentences while the reference answers tended to be at most a few words. On the question "How many trains are on the SHOP W. daily except April 19, 1891?", the generated answer was "There are four trains (Trains 1, 2, 4 and 4) on the SHOP W. daily except April 19, 1891." while the reference answer was a single character: "4". This example yielded precision and recall scores of 0.048 and 1.000 respectively, while cosine similarity was 0.274. This hypothesis is further supported by the improved mean recall score, showing that the generated answers typically covered more tokens in the reference answers.

Despite the poor QAS performances on the Sarawak Gazette QA Dataset, they still match well with previous results for a QAS on historical book question answering (Thapliyal, 2023). This study's QAS achieved 55% higher mean precision than the previous state-of-the-art of 0.316 mean precision, as well as 42% higher mean recall than the previous result of 0.304. However, this study's QAS attained 11% less cosine similarity compared to the other study, achieving only a result of 0.678 compared to the previous state-of-the-art of 0.763 mean cosine similarity.

The value of the average mean evaluation score was found to be 2.9, showing that this method yields high-quality data. This is possibly due to LLMs having been adjusted to be digital assistants in a modern context, which leads to the generated data having modern terminology that is easier to understand, instead of reusing the same language as in the Sarawak Gazette. However, a substantial disagreement between annotators' scores is evident. A likely reason for this is that the annotators had various levels of mastery in subjects other than Sarawakian history, such as English literacy, that may have caused a few annotators to be more skeptical of the generated content than other annotators. This is evident from a comment given by Annotator 7, who stated that on one of the dataset pairs, "the answer should be in past tense, and the format is strange".

Despite this disagreement, the dataset generation technique of this study still shows a clear advantage over previous methods. The average mean evaluation score on this study's dataset is much higher than any of the average mean evaluation scores obtained by a previous study (Arcan et al, 2022). Their four datasets only attained mean evaluation scores ranging between 1.46 to 2.40, whereas this study achieved a score of 2.9, which is 0.5 points better than their best-performing dataset generation method and is also only 0.1 points from the maximum possible score of 3.0 for this metric (Coughlin, 2003).

4 DISCUSSION

This section presents the implications of this study's QAS and dataset creation methodology, as well as address this study's limitations. The main goals of this study were to measure the performance of the question answering system and to develop a dataset to evaluate a question answering chatbot for the Sarawak Gazette.

This study's QAS can generate plausible answers to questions relating to the Sarawak Gazette. The performance values of this study's QAS on the Sarawak Gazette are consistent with previous results for historical book question answering (Thapliyal, 2023). Apart from a slight discrepancy, this study's results provide convincing evidence that a QAS for the Sarawak Gazette can be created using chatbot technology. However, these results could also be interpreted as the QAS being more concerned with just repeating the words in the text rather than preserving the meaning of them. These results thus need to be interpreted carefully.

The performance of the QAS on the *ChroniclingAmericaQA* dataset was rather disappointing. This apparent lack of correlation between the two sets of results can be explained by the different structures of the reference answers in both datasets. This study's Sarawak Gazette dataset expects complete sentences, while the *ChroniclingAmericaQA* dataset expects singular phrases not formatted into proper sentences, thus this deficient performance was not entirely unexpected. In fact, the authors of this study argue that well-formed sentences engage with the end users of the QAS more than single phrases. Nevertheless, there is evidence to suggest the hypothesis that another Sarawak Gazette question answering dataset containing briefer reference answers may yield a QAS that would perform better on other question answering benchmarks.

The data suggest that this study's dataset is of high quality, higher than any of Arcan et al's (2022) datasets. As anticipated by Arcan et al (2022), this study shows that the usage of a pre-trained language model does yield higher quality questions and reference answers than their template-based approach. It seems that using language models for automatic question generation can lead to a higher quality question answering dataset than other automated methods. This might even generate text data that contain simpler and more modern words and phrases that are easier to understand.

It is plausible that the limitations of this study may have influenced the results. The first is that the testing dataset's annotators are no more than three degrees separated from this study's authors. It cannot be ruled out that this has resulted in unintended bias. However, performing a truly robust question verification process is out of scope for this small study. The additional source of error is that RAG over an LLM is performed instead of fine-tuning, due to the time and resource constraints of this study. It is anticipated that LLM fine-tuning would yield more favorable results, as fine-tuning directly modifies the language model's parameters, which allows more drastic changes to the LLM's behavior (Ling et al, 2023).

5 CONCLUSION

The question answering system for the Sarawak Gazette performed fairly well. This research demonstrates that chatbot technology and this study's question answering dataset creation method are viable components for building question answering systems over the Sarawak Gazette, which can also offer the added benefit of the generated answers being more polite and human-like. The source code of this QAS has been released on GitHub (Yusuf, 2024).

This could conceivably lead to historians and the public getting the information they seek from the Sarawak Gazette more quickly and directly. In fact, these methods towards building such systems are applicable to any collection of historical Malaysian gazettes if most of the content is in English.

The present study has only examined the use of English chatbots to answer user questions via retrieval augmented generation. It is recommended that further research should be undertaken in the following areas:

- Use a chatbot that has been pre-trained with more languages that are commonly spoken in Malaysia, such as MaLLaM (Zolkepli & Nooh, 2024), then apply this study's techniques over historical Malaysian documents written in Malay.
- Better integrate an LLM as the retriever component of the question answering system by fine-tuning the model instead of using retrieval augmented generation. Performing this research should also involve the use of a larger question answering dataset involving thousands of examples.

Other than the above, the future work of this study's authors will explore the use of other IR techniques such as semantic web technologies to act as the retriever component of the QAS instead of relying on the cosine similarity metric. Further studies that take better selection of annotators into account, such as through random sampling, will also need to be undertaken.

ADDITIONAL INFORMATION AND DECLARATIONS

Acknowledgments: The authors acknowledged the financial support from the Ministry of Higher Education Malaysia through Fundamental Research Grant Scheme (FRGS) FRGS/1/2020/WAB01/UNIMAS/03/1. The authors would like to express their appreciation to Universiti Malaysia Sarawak for supporting this publication and extend their gratitude to the validators who contributed to the study by providing their valuable feedback.

Conflict of Interests: The authors declare no conflict of interest.

Author Contributions: Y.L.b.Y.: Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – Original Draft. S.b.S.: Conceptualization, Project Administration, Resources, Supervision, Writing – Reviewing and Editing.

Statement on the Use of Artificial Intelligence Tools: The authors declare that they didn't use artificial intelligence tools for text or other media generation in this article.

Data Availability: The data that support the findings of this study are available from the corresponding author and on GitHub (Yusuf, 2024).

REFERENCES

- Adamopoulou, E., & Moussiades, L. (2020). An overview of chatbot technology. In *IFIP international conference on artificial intelligence applications and innovations* (pp. 373–383). Springer. https://doi.org/10.1007/978-3-030-49186-4_31
- Allam, A. M. N., & Haggag, M. H. (2012). The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences*, 2(3), 1–12.
- Anand, Y., Nussbaum, Z., Treat, A., Miller, A., Guo, R., Schmidt, B., GPT4All Community, Duderstadt, B., Schmidt, B., & Mulyar, A. (2023). GPT4All: An Ecosystem of Open Source Compressed Language Models. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software*, (pp. 59–64). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2023.nlp4dh-1.7>
- Arcan, M., O'Halloran, R., Robin, C., & Buitelaar, P. (2022). Towards Bootstrapping a Chatbot on Industrial Heritage through Term and Relation Extraction. In *Proceedings of the 2nd International Workshop on Natural Language Processing for Digital Humanities* (pp. 108–122). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2022.nlp4dh-1.15>
- Auli, M., & Gao, J. (2014). Decoder integration and expected BLEU training for recurrent neural network language models. <https://www.microsoft.com/en-us/research/publication/decoder-integration-and-expected-bleu-training-for-recurrent-neural-network-language-models/>

- Bengio, Y., Ducharme, R., & Vincent, P. (2000). A neural probabilistic language model. In *Advances in neural information processing systems*. NeurIPS. https://proceedings.neurips.cc/paper_files/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf
- Bingham, A. (2010). The digitization of newspaper archives: Opportunities and challenges for historians. *Twentieth Century British History*, 21(2), 225–231. <https://doi.org/10.1093/tcbh/hwq007>
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877–1901. <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html>
- Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T. & Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*. <https://doi.org/10.48550/arXiv.2303.12712>
- Casas, J., Tricot, M. O., Khaled, O. A., Mugellini, E., & Cudré-Mauroux, P. (2020). Trends & methods in chatbot evaluation. In *Companion Publication of the 2020 International Conference on Multimodal Interaction* (pp. 280–286). ACM. <https://doi.org/10.1145/3395035.3425319>
- ChatGPT. (2024). ChatGPT. OpenAI. <https://openai.com/chatgpt>
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., & Fiedel, N. (2023). PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(1), 11324–11436.
- Colby, K. M., Weber, S., & Hilf, F. D. (1971). Artificial paranoia. *Artificial intelligence*, 2(1), 1–25. [https://doi.org/10.1016/0004-3702\(71\)90002-6](https://doi.org/10.1016/0004-3702(71)90002-6)
- Coughlin, D. (2003). Correlating automated and human assessments of machine translation quality. In *Proceedings of Machine Translation Summit IX: Papers*. ACL.
- de Mulder, W., Bethard, S., & Moens, M. F. (2015). A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech & Language*, 30(1), 61–98. <https://doi.org/10.1016/j.csl.2014.09.005>
- Deutsch, D., Bedrax-Weiss, T., & Roth, D. (2021). Towards question-answering as an automatic metric for evaluating the content quality of a summary. *Transactions of the Association for Computational Linguistics*, 9, 774–789. https://doi.org/10.1162/tacl_a_00397
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. <https://doi.org/10.48550/arXiv.1810.04805>
- Dong, L., Wei, F., Zhou, M., & Xu, K. (2015). Question answering over freebase with multi-column convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, (pp. 260–269). ACL.
- Drori, I., Zhang, S., Shuttleworth, R., Tang, L., Lu, A., Ke, E., Liu, K., Chen, L., Tran, S., Cheng, N., Wang, R., Singh, N., Patti, T. L., Lynch, J., Shporer, A., Verma, N., Wu, E., & Strang, G. (2022). A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level. *Proceedings of the National Academy of Sciences*, 119(32), e2123433119. <https://doi.org/10.1073/pnas.2123433119>
- Dumais, S., Banko, M., Brill, E., Lin, J., & Ng, A. (2002). Web question answering: Is more always better?. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, (pp. 291–298). ACM. <https://doi.org/10.1145/564376.564428>
- Embeddings. (2024). Embeddings. OpenAI. <https://platform.openai.com/docs/guides/embeddings>
- Ferret, O., Grau, B., Hurault-Plantet, M., Illouz, G., Monceaux, L., Robba, I., & Vilnat, A. (2001). Finding an answer based on the recognition of the question focus. In *Proceedings of The Tenth Text REtrieval Conference, TREC*. <https://hal.science/hal-02458025/>
- Fong, T., & Ranaivo-Malançon, B. (2014). Using TEI XML schema to encode the structures of Sarawak Gazette. *International Journal of Social Science and Humanity*, 5(10), 855–859. <https://doi.org/10.7763/ijssh.2015.v5.569>
- Gao, J., & Lin, C. Y. (2004). Introduction to the special issue on statistical language modeling. *ACM Transactions on Asian Language Information Processing*, 3(2), 87–93. <https://doi.org/10.1145/1034780.1034781>
- Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57, 345–420. <https://doi.org/10.1613/jair.4992>
- GPT4All. (2024). GPT4All Python SDK. Nomic. https://docs.gpt4all.io/gpt4all_python/home.html
- Haller, E., & Rebedea, T. (2013). Designing a chat-bot that simulates an historical figure. In *2013 19th international conference on control systems and computer science*, (pp. 582–589). IEEE. <https://doi.org/10.1109/CSCS.2013.85>
- Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. D. L., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., & Sifre, L. (2022). Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*. <https://doi.org/10.48550/arXiv.2203.15556>
- Hovy, E. H., Gerber, L., Hermjakob, U., Junk, M., & Lin, C. Y. (2000). Question Answering in Webclopedia. In *Text REtrieval Conference*, (pp. 53–56). NIST.

- Introduction.** (2024). Introduction. *LangChain*. https://python.langchain.com/docs/get_started/introduction/
- Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. D. L., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., & Sayed, W. E.** (2023). Mistral 7B. *arXiv preprint arXiv:2310.06825*. <https://doi.org/10.48550/arXiv.2310.06825>
- Jin, H., Zhang, Y., Meng, D., Wang, J., & Tan, J.** (2024). A Comprehensive Survey on Process-Oriented Automatic Text Summarization with Exploration of LLM-Based Methods. *arXiv preprint arXiv:2403.02901*. <https://doi.org/10.48550/arXiv.2403.02901>
- Joshi, V., Peters, M., & Hopkins, M.** (2018). Extending a parser to distant domains using a few dozen partially annotated examples. *arXiv preprint arXiv:1805.06556*. <https://doi.org/10.48550/arXiv.1805.06556>
- Jurafsky, D., & Martin, J. H.** (2024). *Speech and Language Processing*. <https://web.stanford.edu/~jurafsky/slp3/>
- Kalla, D., Smith, N., Samaah, F., & Kuraku, S.** (2023). Study and Analysis of ChatGPT and its Impact on Different Fields of Study. *International Journal of Innovative Science and Research Technology*, 8(3), 827–833.
- Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J. & Amodei, D.** (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*. <https://doi.org/10.48550/arXiv.2001.08361>
- Kokubu, T., Sakai, T., Saito, Y., Tsutsui, H., Manabe, T., Koyama, M., & Fujii, H.** (2005). The Relationship between Answer Ranking and User Satisfaction in a Question Answering System. In *Proceedings of NTCIR-5 Workshop Meeting*, (pp. 1–8). NII.
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R.** (2019). ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*. <https://doi.org/10.48550/arXiv.1909.11942>
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L.** (2019). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*. <https://doi.org/10.48550/arXiv.1910.13461>
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D.** (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *34th Conference on Neural Information Processing Systems*, (pp. 9459–9474). NeurIPS.
- Ling, C., Zhao, X., Lu, J., Deng, C., Zheng, C., Wang, J., Chowdhury, T., Li, Y., Cui, H., Zhang, X., Zhao, T., Panalkar, A., Mehta, D., Pasquali, S., Cheng, W., Wang, H., Liu, Y., Chen, Z., Chen, H., White, C., Gu, Q., Pei, J., Yang, C., & Zhao, L.** (2023). Domain specialization as the key to make large language models disruptive: A comprehensive survey. *arXiv preprint arXiv:2305.18703*. <https://doi.org/10.48550/arXiv.2305.18703>
- Liu, C. W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., & Pineau, J.** (2016). How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*. <https://doi.org/10.48550/arXiv.1603.08023>
- Liu, X., & Croft, W. B.** (2005). Statistical language modeling for information retrieval. *Annual Review Information Science, Statistics and Probability*, 39(1), 1–31. <https://doi.org/10.21236/ADA440321>
- Liu, Y., Moosavi, N. S., & Lin, C.** (2023). LLMs as narcissistic evaluators: When ego inflates evaluation scores. *arXiv preprint arXiv:2311.09766*. <https://doi.org/10.48550/arXiv.2311.09766>
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V.** (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*. <https://doi.org/10.48550/arXiv.1907.11692>
- Luger, E., & Sellen, A.** (2016). "Like Having a Really Bad PA": The Gulf between User Expectation and Experience of Conversational Agents. In *Proceedings of the 2016 CHI conference on human factors in computing systems* (pp. 5286–5297). ACM. <https://doi.org/10.1145/2858036.2858288>
- Marietto, M. D. G. B., de Aguiar, R. V., Barbosa, G. D. O., Botelho, W. T., Pimentel, E., Franca, R. D. S., & da Silva, V. L.** (2013). Artificial intelligence markup language: a brief tutorial. *arXiv preprint arXiv:1307.3091*. <https://doi.org/10.48550/arXiv.1307.3091>
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J.** (2013). Distributed representations of words and phrases and their compositionality. In *NIPS'13: Proceedings of the 27th International Conference on Neural Information Processing Systems – Volume 2*, (pp. 3111–3119). NIPS.
- Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J.** (2024). Large language models: A survey. *arXiv preprint arXiv:2402.06196*. <https://doi.org/10.48550/arXiv.2402.06196>
- Mishra, A., & Jain, S. K.** (2016). A survey on question answering systems with classification. *Journal of King Saud University-Computer and Information Sciences*, 28(3), 345–361. <https://doi.org/10.1016/j.jksuci.2014.10.007>
- Mistral AI Team.** (2023). Mixtral of Experts. *Mistral AI*. <https://mistral.ai/news/mixtral-of-experts/>
- Nomic AI.** (2024). Run Large Language Models Locally. *Nomic*. <https://www.nomic.ai/gpt4all>
- Nor Azizan, N. A. B., Saee, S. B., & Yusof, M. A. B.** (2023). Discovering Popular Topics of Sarawak Gazette (SaGa) from Twitter Using Deep Learning. In *International Conference on Soft Computing in Data Science* (pp. 178–192). Springer. https://doi.org/10.1007/978-981-99-0405-1_13
- Obligations of the GPL and LGPL.** (2024). Obligations of the GPL and LGPL. *The Qt Company*. <https://www.qt.io/licensing/open-source-lgpl-obligations>
- Ojokoh, B., & Adebisi, E.** (2018). A review of question answering systems. *Journal of Web Engineering*, 17(8), 717–758. <https://doi.org/10.13052/jwe1540-9589.1785>
- Ong, C. S., Day, M. Y., & Hsu, W. L.** (2009). The measurement of user satisfaction with question answering systems. *Information & Management*, 46(7), 397–403. <https://doi.org/10.1016/j.im.2009.07.004>
- OpenAI et al.** (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*. <https://doi.org/10.48550/arXiv.2303.08774>

- Paolucci, M., Kawamura, T., Payne, T. R., & Sycara, K. (2002). Semantic matching of web services capabilities. In *The Semantic Web—ISWC 2002: First International Semantic Web Conference*, (pp. 333–347). Springer. https://doi.org/10.1007/3-540-48005-6_26
- Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311–318). ACL.
- Piryani, B., Mozafari, J., & Jatowt, A. (2024). ChroniclingAmericaQA: A Large-scale Question Answering Dataset based on Historical American Newspaper Pages. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2038–2048). ACM. <https://doi.org/10.1145/3626772.3657891>
- Pricing. (2024). Pricing. OpenAI. <https://openai.com/pricing>
- Pustaka Negeri Sarawak. (2013). e-Sarawak Gazette. WhiteHornbill. <https://www.pustaka-sarawak.com/gazette/home.php>
- Q&A with RAG. (2024). Q&A with RAG. LangChain. https://python.langchain.com/docs/use_cases/question_answering/
- Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. <https://www.mikecaptain.com/resources/pdf/GPT-1.pdf>
- Ramli, F., Ranaivo-Malançon, B., Chua, S., & Mohammad, M. S. (2017). Comparative Studies of Ontologies on Sarawak Gazette. *Journal of Telecommunication, Electronic and Computer Engineering*, 9, 61–65.
- Retrieval Augmented Generation. (2024). Retrieval Augmented Generation. Databricks. <https://www.databricks.com/glossary/retrieval-augmented-generation-rag>
- Rosita, M. O., Fatihah, R., Nazri, K. M., Yeo, A. W., & Tan, D. Y. (2010). Cultural Heritage Knowledge Discovery: An Exploratory Study of the Sarawak Gazette. In *2nd Semantic Technology and Knowledge Engineering Conference* (pp. 20–27). UNIMAS.
- rlm. (2023). rag-prompt. LangChain. <https://smith.langchain.com/hub/rlm/rag-prompt>
- Roy, P. K., Saumya, S., Singh, J. P., Banerjee, S., & Gutub, A. (2023). Analysis of community question-answering issues via machine learning and deep learning: State-of-the-art review. *CAAI Transactions on Intelligence Technology*, 8(1), 95–117. <https://doi.org/10.1049/cit2.12081>
- Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*. <https://doi.org/10.48550/arXiv.1910.01108>
- Santos, J., Rodrigues, J. J., Casal, J., Saleem, K., & Denisov, V. (2016). Intelligent personal assistants based on internet of things approaches. *IEEE Systems Journal*, 12(2), 1793–1802. <https://doi.org/10.1109/JSYST.2016.2555292>
- Shanahan, M. (2024). Talking about large language models. *Communications of the ACM*, 67(2), 68–79. <https://doi.org/10.1145/3624724>
- Shawar, A., & Atwell, E. S. (2005). A chatbot system as a tool to animate a corpus. *Journal: International Computer Archive of Modern and Medieval English Journal*, 29, 5–24.
- Silva, A. D. B., Gomes, M. M., da Costa, C. A., da Rosa Righi, R., Barbosa, J. L. V., Pessin, G., Doncker, G. D. & Federizzi, G. (2020). Intelligent personal assistants: A systematic literature review. *Expert Systems with Applications*, 147, 113193. <https://doi.org/10.1016/j.eswa.2020.113193>
- Tan, Y., Min, D., Li, Y., Li, W., Hu, N., Chen, Y., & Qi, G. (2023). Can ChatGPT replace traditional KBQA models? An in-depth analysis of the question answering performance of the GPT LLM family. In *International Semantic Web Conference* (pp. 348–367). Springer Nature. https://doi.org/10.1007/978-3-031-47240-4_19
- Taye, M. M. (2010). Understanding semantic web and ontologies: Theory and applications. *arXiv preprint arXiv:1006.4567*. <https://doi.org/10.48550/arXiv.1006.4567>
- Tebbe, J-P. (2024). Enhancing Historical Research with RAG Chatbots. GitHub. <https://github.com/Thukyd/azure-search-openai-hackathon>
- Text embedding models. (2024). Text embedding models. LangChain. https://python.langchain.com/v0.1/docs/modules/data_connection/text_embedding/
- Thapliyal, H. (2023). *Unveiling the Past: AI-Powered Historical Book Question Answering*. Doctoral dissertation. Swiss School of Business and Management Geneva.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*. <https://doi.org/10.48550/arXiv.2302.13971>
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236), 433–460. <https://doi.org/10.1093/mind/LIX.236.433>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. In *31st Conference on Neural Information Processing Systems*, (NIPS 2017). NISP.
- Wallace, R. S. (2009). The anatomy of A.L.I.C.E. In Epstein, R., Roberts, G., Beber, G. (eds) *Parsing the Turing Test*, (pp. 181–210). Springer. https://doi.org/10.1007/978-1-4020-6710-5_13
- Wang, P., Li, L., Chen, L., Cai, Z., Zhu, D., Lin, B., Cao, Y., Liu, Q., Liu, T., & Sui, Z. (2024). Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, (pp. 9440–9450). ACL. <https://doi.org/10.18653/v1/2024.acl-long.511>
- Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36–45. <https://doi.org/10.1145/365153.365168>
- White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with ChatGPT. *arXiv preprint arXiv:2302.11382*. <https://doi.org/10.48550/arXiv.2302.11382>
- Yusuf, Y. L. (2024). A Question and Answering System for the Sarawak Gazette Using Chatbot Technology. GitHub. <https://github.com/TelluricSpeck17101/SarawakGazetteQAS>

- Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J. Y., & Wen, J. R. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*. <https://doi.org/10.48550/arXiv.2303.18223>
- Zhai, C. (2008). Statistical language models for information retrieval a critical review. *Foundations and Trends in Information Retrieval*, 2(3), 137–213. <https://doi.org/10.1561/15000000008>
- Zheng, L., Chiang, W. L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., & Stoica, I. (2023). Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *NIPS '23: Proceedings of the 37th International Conference on Neural Information Processing Systems*, (pp. 46595–46623). NeurIPS.
- Zolkepli, H., & Nooh, K. (2024). MaLLaM. *Mesolitica*. <https://mesolitica.com/mallam>