

Bachelor's Programme in Science and Technology

Next Generation of Intelligent Chatbots

Review of Approaches to Intelligent Conversational Agents

Ali Amaan

**Bachelor's Thesis
2023**



This work is licensed under a Creative Commons Attribution 4.0 International License.
Copyright © 2023 Ali Amaan

Author Ali Amaan		
Title of thesis Review of Approaches to Intelligent Conversational Agents		
Programme Bachelor’s Programme in Science and Technology		
Major Digital Systems and Design		Code: ELEC 3056
Thesis supervisor Martin Andraud		
Thesis advisor Salu Ylirisku		
Date 26.04.2023	Number of pages 52	Language English

Abstract

The fusion of AI with conversational agents (used interchangeably with chatbots) has recently been a prevalent domain due to the advent of ChatGPT, GPT 4, and other natural language generation models that perform exceptionally well. This thesis presents a comprehensive literature review to determine the current state of research on conversational agents and chatbot applications utilizing natural language processing (NLP) models. By examining scholarly works from August 2016 to April 2023, the study identifies limitations in the field and proposes a potential approach for the advancement of chatbot technology. The insights derived from this review are intended to guide researchers, experts, and developers in creating the next generation of intelligent chatbots and ultimately contribute to developing early Artificial General Intelligence (AGI) systems.

The review covers an extensive range of areas, including existing chatbot technologies, generative large language models (LLMs), and their limitations. It also provides a detailed analysis of prominent LLMs such as the GPT series, BERT, T5, RoBERTa, and Sentence Transformers models. Then this foundational knowledge was bridged with conversational agents. This thesis introduces a proposed approach for the next generation of chatbots, incorporating crucial process blocks such as input pre-processing, prompt engineering, contextualization, chain-of-thought prompting, fine-tuning, countering hallucinations and inconsistencies, and a reinforcement learning-based feedback system. In conclusion, this literature review thoroughly examines the advancements and challenges in conversational agents and significant LLM technologies with a proposed approach incorporating the modules discussed earlier. The insights, recommendations, approaches, and discussions in this thesis serve as valuable resources for stakeholders in prospective AI-fused conversational agent development.

Keywords Natural Language Processing (NLP); Chatbot; Conversational Agents; Artificial Intelligence; Prompt Engineering; Next Generation Intelligent Chatbots; GPT; AGI.

Preface

This bachelor's thesis was conducted as a part of the Bachelor's Degree Programme in Science and Technology at Aalto University. I want to thank my advisor, Salu Ylirisku, for his tremendous support, encouragement, and valuable feedback throughout the process. I would also like to thank Marice Forget for enabling me to stay on track and helping me get my thesis aptly in shape. Lastly, I would like to thank Martin Andraud for his supervision support.

This thesis is dedicated to Islam, my way of life, which has always pushed me to strive for the best. I want to devote this feat to Prophet Muhammad ﷺ, His Companions, my maternal grandfather Khursheed Ahmed, my late paternal grandfather Muhammad Rashid, my parents, my younger siblings, especially Ahmed Wali, and all my loving and supporting family.

A handwritten signature in black ink, appearing to read 'Ali Amaan', with a stylized flourish at the end.

Ali Amaan

26th April 2023

Otaniemi, Espoo, Finland.

Table of Contents

Abstract	3
Preface.....	4
List of Figures	7
List of Tables.....	8
List of Abbreviations.....	9
1 Introduction	10
1.1 Motivation.....	10
1.2 Overview	11
2 Literature Review of Existing Technologies	12
2.1 Chatbots or Conversational Agents.....	12
2.1.1 Approaches.....	12
2.1.2 High-Level Processes of ML-Based Chatbots.....	15
2.2 Generative LLM and NLP Models.....	16
2.2.1 Comparison of Select Generative LLM Generative Models	17
2.2.1.1 Generative Pre-trained Transformer (GPT) Models by OpenAI	18
2.2.1.2 Bidirectional Encoder Representations from Transformer (BERT) by Google	19
2.2.1.3 Text-to-Text Transfer Transformer (T5) by Google	19
2.2.1.4 Robustly-optimized BERT approach (RoBERTa) by Facebook	20
2.2.1.5 Sentence Transformer Models by Hugging Face	20
2.2.1.6 Summary of LLMs	21
2.2.2 Selection of NLP Models for Use in Chatbots	21
2.2.3 Domain-Specific Models.....	23
2.2.4 Limitations of LLMs	24
3 A Proposed Approach for the Next Generation of Chatbots	26
3.1 A Review of Approaches to Chatbot Intelligence	26
3.2 Crucial Process Blocks	27
3.2.1 Input Preprocessing.....	27
3.2.2 Prompt Engineering and Contextualization.....	28
3.2.2.1 Behavior Priming.....	29
3.2.2.2 Zero-Shot and Few-Shot Prompting	30
3.2.2.3 Search & Retrieval from Connected Knowledge Bases.....	31
3.2.2.4 Maintaining Conversation History.....	33
3.2.3 Chain of Thought Prompting & Prompt Chaining	34

3.2.4	Fine-Tuning	36
3.2.5	Countering Hallucinations & Inconsistencies	37
3.2.6	RL and Feedback System	38
3.3	Example Proposed Approach	39
4	Discussion	40
5	Conclusion.....	42
	References	43

List of Figures

FIGURE 1: MIND MAP OF VARIOUS ASPECTS AND PROPERTIES LINKED WITH CHATBOTS (SINGH & BENIWAL, 2022)	12
FIGURE 2: RULE-BASED FLOWCHART FOR CHATBOT LOGIC (D'AVILA, 2018)	13
FIGURE 3: GENERAL CHATBOT ARCHITECTURE (ADAMOPOULOU & MOUSSIADES, 2020B)	15
FIGURE 4: PROMPT-BASED SHOT LEARNING CONCEPT (ZHOU ET AL., 2022)	30
FIGURE 5: ARCHITECTURE MODEL FOR A SEMANTIC SEARCH AND RETRIEVAL SYSTEM (OGUNDEPO, 2022) ..	31
FIGURE 6: RETRIEVAL TEXT DATA INJECTED INTO THE LLM PROMPT	33
FIGURE 7: EFFECT ON LLM RESPONSE WITH AND WITHOUT CONVERSATIONAL MEMORY (PINECONE, 2023A)	34
FIGURE 8: CHAT HISTORY INCORPORATION INTO THE LLM PROMPT	34
FIGURE 9: CHAIN OF THOUGHT PROMPTING EFFECT USING ONE SHOT PROMPT (WEI ET AL., 2022).....	35
FIGURE 10: PROMPT CHAINING EXAMPLE FOR A MUSIC CHATBOT (WU ET AL., 2022)	36
FIGURE 11: SELF-REFINE ALGORITHM OF AUTOMATIC FEEDBACK LOOP (MADAAN ET AL., 2023)	39
FIGURE 12: CONCEPT MAP OF THE PROPOSED LLM-BASED INTELLIGENT CHATBOT APPROACH	39
FIGURE 13: THE RISK AND OPPORTUNITIES OF LLMs WITH RESPECT TO THE ENVIRONMENT (RILLIG ET AL., 2023)	41

List of Tables

TABLE 1: AIML SYNTAX EXAMPLE	14
TABLE 2: OVERVIEW OF NLP BENCHMARK METRICS FOR PERFORMANCE EVALUATION	18
TABLE 3: A SUMMARY OF THE DISCUSSED LLM MODELS FOR POTENTIAL USE IN THE CHATBOT IMPLEMENTATION APPROACH PROPOSAL.....	21
TABLE 4: PERFORMANCE OF LLMs ON ACADEMIC BENCHMARKS AS REPORTED IN THE TECHNICAL REPORT BY OPENAI (2023C) AS OF 14TH MARCH 2023. COMPARISON IS BETWEEN GPT-4, GPT-3.5, THE BEST SOTA LANGUAGE MODELS.....	22
TABLE 5: PROMPT PATTERNS AND CATEGORIES (WHITE ET AL., 2023)	28
TABLE 6: EXPLANATIONS OF VECTOR SIMILARITY METRICS (SCHWABER-COHEN & PINECONE, 2023)	32

List of Abbreviations

AI	Artificial intelligence
ML	Machine learning
API	Application programming interface
NLP	Natural language processing
LLM	Large language model
NER	Named entity recognition
NLU	Natural language understanding
NLG	Natural language generation
GPT	Generative pre-trained transformer
BERT	Bidirectional encoder representations from transformers
RoBERTa	Robustly-optimized BERT approach
HyDE	Hypothetical document embeddings
RL	Reinforcement learning
HMMs	Hidden Markov models
SVMs	Support vector machines
AGI	Artificial general intelligence
SOTA	State-of-the-art

1 Introduction

Artificial Intelligence (AI) has catalyzed advancements in multiple fields of human society, ranging from healthcare to finance to national security (West & Allen, 2018). One such area of application is chatbots or conversational agents. This thesis focuses on the literature review of chatbot approaches and large language AI models to propose an approach for the next generation of chatbots.

The AI sub-module that intersects with language and linguistics is ‘Natural Language Processing (NLP)’. NLP is a subfield of AI that focuses on enabling computers to contextualize, interpret, search, or generate natural-sounding language. It involves extensive computational techniques to process large amounts of natural language data, such as text and speech, to train an AI model (Van Dis et al., 2023). NLP has attracted increased interest in recent years, enabling the development of intelligent models that can understand and respond to human queries in a natural and conversational manner. A module of NLP is the text completion module. The recent advances in the performance of text completion modules are close to human conversational level performance such that there is an unnoticeable difference between an AI-generated response and a human response. An increasingly popular example is the conversational AI model used in the web application ChatGPT and GPT-4 by OpenAI (Thorp, 2023).

Subsequently, one of the most popular applications of NLP models is integration in chatbots, which are computer programs designed to simulate human-like conversations with users. The areas of implementation of such conversations include customer service and feedback collection in banks (Venkat, 2020), customer handling in the food industry (Gilliland, 2016), and e-commerce (Acquire.io, 2022). They provide a convenient application of NLP models for real-world users to interact with computer systems in a practical scenario without requiring specialized training, case scenarios for every situation, or technical expertise while improving efficiency measured through key performance indicators.

1.1 Motivation

The aim of this thesis is to determine the current state of research on the latest techniques for developing conversational agents and chatbot applications that utilize NLP models. This is accomplished by conducting a state-of-the-art review of scholarly works, majorly from August 2016 till April 2023, to determine their limitations and provide a potential comprehensive approach proposal for advancement in the field. Overall, the review provides insights that can help researchers, experts, and developers to develop the next generation of intelligent chatbots and potentially an early version of a functional AGI system – a term popularized by Ben Goertzel, Shane Legg, and Mark Gubrud (Goertzel, 2011), who used it to differentiate a general form of AI close to humans from narrow AI capable of specific domains or tasks.

However, the development of NLP-based chatbots is still evolving with novel approaches. The development involves several aspects, including understanding user

intent, extracting relevant information from possible external knowledge bases, generating appropriately tuned responses, prompt engineering, and incorporating contextualization by recording chat history. These aspects require various NLP techniques, such as natural language understanding (NLU), natural language generation (NLG), and other software technologies not particularly native to AI. The techniques used for developing chatbots have their own natural and imposed limitations that hinder their ability to handle complex queries and generate up-to-date human-like responses innately.

1.2 Overview

This literature review thesis is structured as follows.

Section 1 overviews, introduces, and discusses the motivation behind this thesis. **Section 2** elaborates on the literary findings in the context of conversational agents and AI models. First, **Section 2.1** describes the current state and high-level overview of conversational agents, including a module-wise breakdown of the process flows involved. Then, **Section 2.2** presents a comparative analysis of the current state of selected AI LLM models alongside their limitations and use cases contextualized for use in chatbots.

Section 3 presents a module-wise breakdown of the potential technologies involved and how LLMs integrate with them to advance intelligent chatbots further. Towards the end of this section, an approach proposal with a simple process flow diagram is also put forth that unites the modules discussed. **Section 4** discusses and reflects on the findings of this thesis, along with directions for future research and consideration for ethical concerns. **Section 5** summarizes and concludes the thesis.

2 Literature Review of Existing Technologies

In **Section 2**, the current state of chatbot systems and NLP are focused on with two primary areas of investigation. First, in **Section 2.1**, "Chatbots or Conversational Agents", various types and classifications of chatbots are explored. Their underlying technologies, development platforms, and applications in diverse domains are also conferred upon. Then, in **Section 2.2**, "Generative LLM and NLP Models," the advancements in large language models (LLMs) and their impact on NLP tasks are examined. This section provides an in-depth analysis of generative models, such as GPT-3, and their roles in enhancing the capabilities of chatbots by enabling them to understand context, generate coherent responses, and create text embeddings. By comprehensively reviewing the existing technologies, a solid foundation for understanding the current landscape of chatbot systems is established.

2.1 Chatbots or Conversational Agents

The pursuit of developing an entity capable of understanding and interacting with its creator has been a long-standing human ambition. However, it was Alan Turing who, in 1950, first introduced the concept of a chatbot when he posited the question, "Can machines think?" (Turing, 2007). Turing's depiction of the conduct of an intelligent machine closely aligns with the modern understanding of a chatbot (Caldarini et al., 2022). **Figure 1** illustrates a mind map of the concepts surrounding chatbots. The thesis will introduce the 'approaches' to current chatbots, which include 'traditional' and 'deep neural-based' methods. The particular focus is on the 'deep neural-based' or AI-based chatbots. With the approaches, the related 'existing frameworks', 'knowledge data' and 'industries' are also discussed.

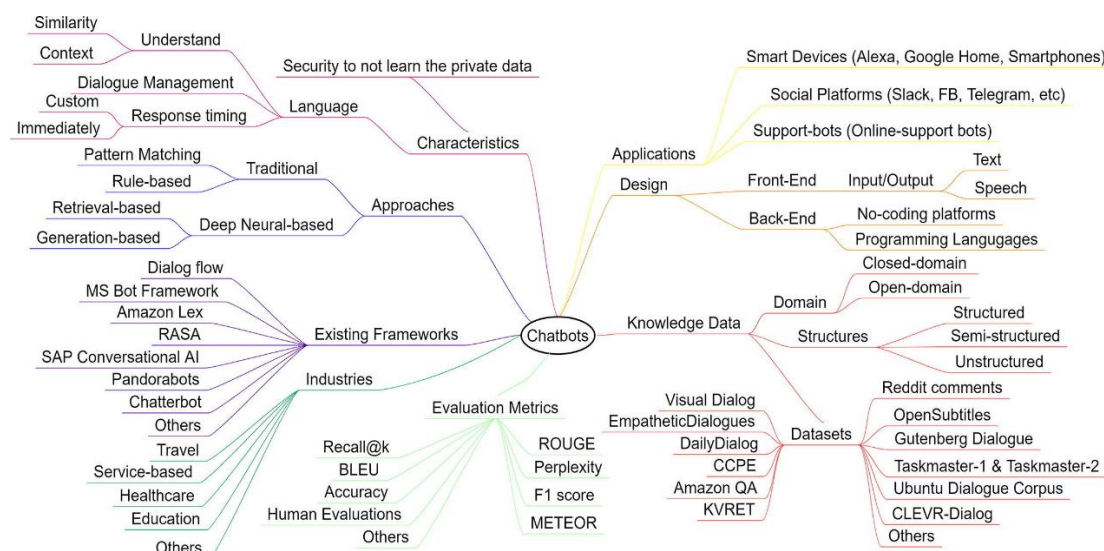


Figure 1: Mind map of various aspects and properties linked with Chatbots (Singh & Beniwal, 2022)

2.1.1 Approaches

Chatbots, also known as conversational agents, have gained significant traction in recent years due to their potential to revolutionize customer service, automate repetitive tasks, and improve overall user experience (Caldarini et al., 2022). The growing sophistication of chatbots is fuelled by the rapid advancements in NLP, particularly LLMs, that have enabled them to understand, interpret, and respond better to natural human language (Caldarini et al., 2022). Accordingly, this literature review section explores the commercial approaches various organizations adopt to develop and deploy chatbots, and the high-level process flows that govern their operation.

The development and deployment of chatbots can be broadly categorized into two commercial approaches: rule-based, pattern matching-based, and machine learning-based.

The rule-based approach (Adamopoulou & Moussiades, 2020b) involves designing chatbots with predefined conversation flows and response templates. This approach relies on a fixed set of rules and decision trees to guide the responses of the chatbot. Rule-based chatbots are easier to build and require minimal data for training. However, they are limited in their ability to understand complex language structures or handle context-dependent queries due to rigid conversation flows. Some rules-based chatbot tools include Manychat (Manychat, 2023), Chatfuel (Chatfuel, 2023), and IBM Watson (IBM, 2023) (Ramesh et al., 2017). The backend logic of these chatbots is similar to the one illustrated in **Figure 2** (D’Avila, 2018).

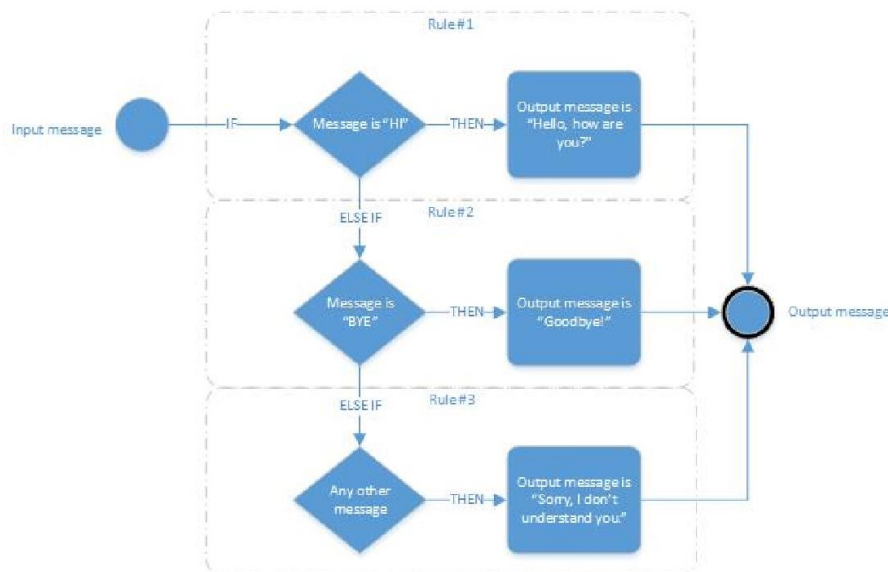


Figure 2: Rule-based flowchart for chatbot logic (D’Avila, 2018)

Akin to a programming flowchart, this approach involves designing extensive conversation flows and rules catering to the appropriate user scenarios. Therefore, they can become very complex with extensive decision trees very quickly.

Pattern matching is another approach which involves specifying predefined outputs and matching them with given inputs according to the keywords characteristics of sentences. This technique is predominantly used in question answering bots. Arti-

ficial Linguistic Internet Computer Entity, also known as A.L.I.C.E., is an example of such a bot, as Wallace (2007) explained. Interestingly, it also won the Loebner prize for ‘the most human computer’ in the Turing Test contests of 2000, 2001, and 2004 (Wallace, 2007). Pattern matching utilizes the artificial intelligence markup language (AIML), a form of eXtensible Markup Language (XML), for defining rules, as shown in **Table 1**.

AIML	
1	<category>
2	<pattern>
3	User input comes here
4	</pattern>
5	<template>
6	Bot response to the input
7	</template>
8	</category>

Table 1: AIML Syntax Example

Machine learning-based chatbots, often powered by LLMs, utilize vast amounts of data to learn and improve their understanding of human language. These chatbots are trained on diverse text datasets, enabling them to handle a broader range of queries and provide more contextually relevant responses. Machine learning-based chatbots are more flexible and adaptive than rule-based chatbots, but they require significant computational resources and substantial amounts of training data to achieve optimal performance.

Classification of chatbots is based on various factors, including the scope of knowledge, the degree of human assistance, and their development platform (Adamopoulou & Moussiades, 2020a). **Figure 1** mentions a few of these aspects.

Regarding the knowledge domain, chatbots are broadly categorized as open-domain or closed-domain. Open-domain chatbots can engage in conversations on a wide range of general topics. In contrast, closed-domain chatbots are specifically designed to address questions within a particular subject area and may not be able to respond effectively to unrelated queries (Nimavat & Champaneria, 2017).

Another criterion for classifying chatbots is the extent of human involvement in their components. Human-aided chatbots rely on input from crowd workers, freelancers, or full-time employees to supplement the logic of the chatbot and address gaps arising from the limitations of fully automated systems. Although human assistance can enhance flexibility and robustness, it is not as scalable as machine-based processing, especially when handling large volumes of user requests (Kucherbaev et al., 2018).

Chatbots can also be differentiated based on the development platforms that they employ. These platforms can be open source, such as RASA (Rasa Community, 2022), or proprietary, as offered by large corporations like Google or IBM. Open-source platforms allow developers greater control over implementation, whereas

closed platforms may act as black boxes with limited customization options. However, closed platforms may provide more immediate access to cutting-edge technologies and benefit from the vast data resources of the parent companies.

It is essential to note that chatbots typically exhibit characteristics from multiple categories, with the specific proportions varying across different implementations.

2.1.2 High-Level Processes of ML-Based Chatbots

The high-level process flows of ML-based chatbots are distributed across five key stages. A concept map of those stages is displayed in Figure 3 by Adamopoulou & Moussiades (2020).

This thesis extracts and contextualizes each of the mentioned high-level processes using the process flow blocks from the general architecture of a chatbot. Later in **Section 3**, these process flow blocks are built upon to propose another approach to chatbots in the wake of powerful AI LLM models.

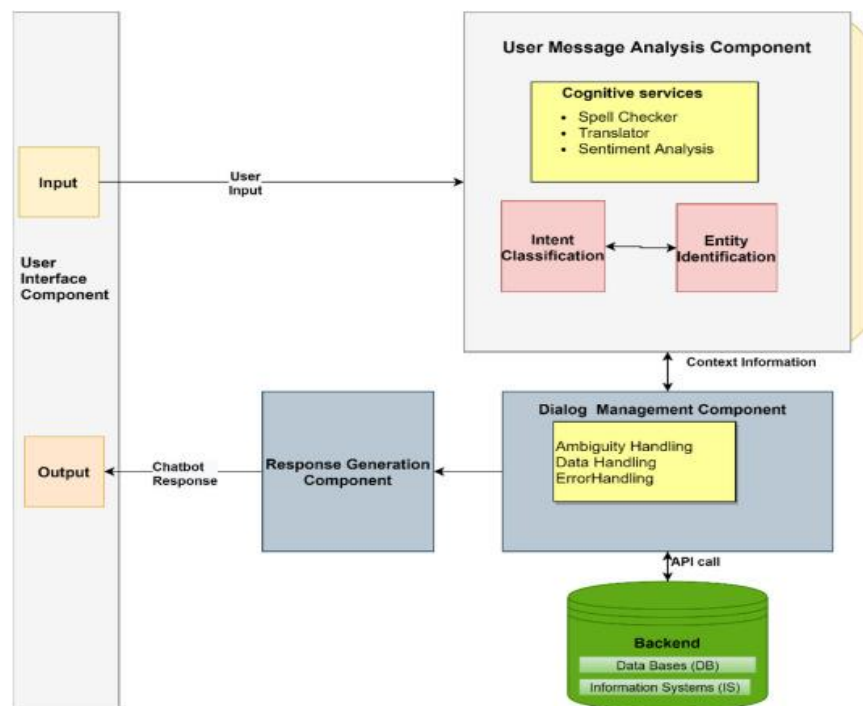


Figure 3: General chatbot architecture (Adamopoulou & Moussiades, 2020b)

Input processing is the first process, as depicted by the input box in the user interface component of **Figure 3**. At this stage, the user inputs a query or a statement, which the chatbot receives as raw text. The chatbot then pre-processes the input by performing tasks, such as tokenization, stemming, and stop-word removal, to convert the text into a format suitable for further processing. Additionally, the chatbot may employ speech recognition techniques if the input is in the form of spoken language.

Intent recognition is the second process illustrated by the intent classification and entity identification modules in **Figure 3**. Once the input is processed, the chatbot must determine the intent of the user or the purpose of the input. In rule-based chatbots, intent recognition is typically achieved through pattern matching or keyword extraction. In contrast, machine learning-based chatbots employ advanced techniques such as word embeddings, deep learning, or LLMs to discern user intent accurately.

Context management is the third process. The dialog management component and the backend databases and information systems components in **Figure 3** display it. It involves maintaining the context of an ongoing conversation and utilizing it to inform the responses of the chatbot. This may include remembering previous interactions or leveraging external data sources to provide a more comprehensive and personalized user experience. Context management is crucial for both rule-based and machine learning-based chatbots, as it enables them to respond intelligently to user queries by taking relevant information from prior interactions and external knowledge bases into account.

Response generation is the fourth process. After identifying the user's intent and maintaining the context of the conversation, the chatbot generates an appropriate response. Rule-based chatbots typically select a predefined response template or execute a specified action based on the matched intent. In contrast, machine learning-based chatbots generate responses by applying AI techniques like sequence-to-sequence models or LLMs, which can create more diverse and contextually accurate responses.

Output delivery is the fifth and final process, where the chatbot delivers its response to the user. In text-based chatbots, the response is typically displayed as a text message. In voice-based chatbots, text-to-speech technology is integrated to convert the generated text response into spoken language. The user then perceives the response of the chatbot and can continue the conversation or end the interaction.

2.2 Generative LLM and NLP Models

NLP is a broad subfield of AI that revolves around the interaction with computers using human languages. It includes developing algorithms and models which enable computers to understand, generate, and interpret human language that is both meaningful and useful. NLP tasks include but are not limited to text classification, sentiment analysis, machine translation, speech recognition, information extraction, information retrieval, and question answering (Shaalán et al., 2017).

On the other hand, LLMs are a specific type of ML models within the NLP domain designed to understand and generate human language. These LLMs are built on top of foundational deep learning architectures, such as recurrent neural networks (RNN), long short-term memory (LSTM) networks, and transformer architecture. First introduced by Vaswani et al. (2017), Transformer model architecture is a widely

used architecture. It refers to a type of neural network that acquires context and meaning by identifying connections within sequential data, such as the words present in a sentence (Lee, 2023). Generative LLMs are a subset of NLP models and have been very successful in setting SOTA performance benchmarks on a wide variety of NLP tasks, which include but are not limited to recognizing, summarizing, translating, predicting, and generating language content (Lee, 2023). Examples of Generative LLMs include Google’s BERT (Bidirectional Encoder Representations from transformers) (Devlin et al., 2018), OpenAI’s GPT-3 (Generative Pre-trained Transformer 3), NVIDIA and Microsoft’s Megatron-Tuning Natural Language Generation, and HuggingFace’s BLOOM (Lee, 2023).

2.2.1 Comparison of Select Generative LLM Generative Models

Generative LLMs have demonstrated significant success in a wide range of NLP tasks, including the development of sophisticated chatbot applications. LLMs, such as OpenAI’s GPT series and Google’s BERT, have exhibited remarkable advancements in generating human-like responses, understanding context, and discerning user intent. The following section highlights a comparative analysis of a few selected generative LLM models suitable for chatbot applications with their strengths, weaknesses, and potential use cases.

Table 2 provides a background of commonly used NLP benchmarks for performance: GLUE, SuperGLUE, SQuAD, LAMBADA, and RACE. These benchmarks are referred to when comparing LLMs.

NLP Benchmark	Description
GLUE (Wang et al., 2018)	General Language Understanding Evaluation (GLUE) is a benchmark suite that evaluates a model on nine NLU tasks: sentiment analysis, acceptability analysis, two paraphrase tasks, similarity task, and three inference tasks. This condenses the evaluation of a model across the nine tasks to a single score.
SuperGLUE (Wang et al., 2019)	SuperGLUE is an extension of the GLUE benchmark, designed in a more diverse and challenging way for the models. It encompasses eight tasks, including natural language inference (CB and RTE), question answering (BoolQ), and coreference resolution (WSC). SuperGLUE also condenses the performance to a single score; however, the tasks are advanced with complex reasoning.
SQuAD (Rajpurkar et al., 2016)	Stanford Question Answering Dataset (SQuAD) is a benchmark that tests the ability of a model to comprehend and answer questions based on given contexts. It comprises of over 100,000 question answer pairs, where the models are evaluated based on their ability to identify the correct answers.
LAMBADA (Paperno et al., 2016)	LAMBADA (LAnguage Modeling Broadened to Account for Dependency and Ambiguity) is a benchmark designed for assessing a model based on how well it predicts the final word in a given sentence using a limited context window. This benchmark

	measures the language modeling skills, capacity to disambiguate and reason about ambiguous language, and ability to comprehend long-range dependencies within sentences.
RACE (Lai et al., 2017)	Reading Comprehension Dataset for English Examination (RACE) is a benchmark that assess a model on reading comprehension tasks specifically assembled for English exams. It contains of over 28,000 passages and 97,000 multiple-choice questions, created from English examinations in schools. RACE tests a model's comprehension analysis, and deductive abilities while mirroring text comprehension problems.

Table 2: Overview of NLP benchmark metrics for performance evaluation

2.2.1.1 Generative Pre-trained Transformer (GPT) Models by OpenAI

Developed by OpenAI, GPT is a cutting-edge LLM series based on the ‘transformer’ architecture. The GPT-3 is trained using 175 billion parameters, whereas GPT-4 is trained using 170 trillion parameters. The latest GPT models have demonstrated impressive performance in a variety of NLP tasks, including text completion, summarization, and translation (OpenAI, 2023c).

In addition to generating coherent and contextually accurate responses, OpenAI's models can also be utilized for creating text embeddings (OpenAI, 2023a). These embeddings are continuous vector representations of textual input, which encapsulate the semantic information of the text in a dense, fixed-dimensional format. Creating such embeddings is crucial for various NLP tasks, including document similarity, information retrieval, and clustering (OpenAI, 2023c).

The GPT series has a number of inherent strengths. Firstly, it is capable of generating highly coherent and contextually relevant responses in chatbot applications (Hagendorff et al., 2022). Secondly, its few-shot learning ability enables GPT-3 to understand tasks and generate responses with minimal examples. This reduces the need for extensive fine-tuning, which is carried out by employing large training data sets containing thousands of data points (Schick & Schütze, 2020). Thirdly, it supports zero-shot and one-shot learning, demonstrating adaptability across various domains (Wei et al., 2021).

On the contrary, it also has some limitations. Firstly, GPT requires substantial computational resources for training and independently hosted deployment due to its large size, which may not be feasible for all organizations (Li, 2023). Additionally, since the model is close source, it cannot be hosted and deployed. However, a usage-based subscription is offered to interact with the model with limited tokenized input through its application programming interface (API) endpoints. Secondly, the model can sometimes produce inaccurate or nonsensical responses, despite appearing plausible. Simply put, it ‘hallucinates’ (Alkaissi & McFarlane, 2023). Thirdly, GPT models may inadvertently generate biased or offensive content based on its training data (Abid et al., 2021).

GPT models are well-suited for chatbot applications that demand highly natural and contextually accurate responses, such as customer support, virtual assistants, and content generation (OpenAI, 2023d).

2.2.1.2 Bidirectional Encoder Representations from Transformer (BERT) by Google

BERT is an influential LLM developed by Google. Unlike GPT, BERT is designed primarily for understanding and representing the input text, making it particularly effective for tasks such as sentiment analysis, named entity recognition, and question answering.

Firstly, the bidirectional training approach of the BERT model allows it to capture the context of a word within a sentence better, improving its understanding of language structure (Dai & Callan, 2019). Secondly, BERT has demonstrated state-of-the-art performance on various NLP benchmarks, such as GLUE and SQuAD, showcasing its effectiveness across multiple tasks (Zhang et al., 2020). Thirdly, BERT can be fine-tuned for specific applications, allowing developers to adapt the model to their unique chatbot requirements (Qasim et al., 2022).

However, BERT also faces some limitations. Firstly, BERT is not inherently designed to generate text, which may limit its effectiveness in generating human-like responses for chatbot applications (Koroteev, 2021). Secondly, similar to GPT-3 and 4, BERT requires significant computational resources for training and independent deployment (Li, 2023) due to its open-source nature. Thirdly, the model may also suffer from biases present in its training data, such as, gender bias (Bhardwaj et al., 2021).

BERT is suitable for chatbot applications that focus on understanding user input and extracting relevant information, such as question-answering systems, sentiment analysis, and information retrieval.

2.2.1.3 Text-to-Text Transfer Transformer (T5) by Google

The T5 is another LLM developed by Google. T5 unifies various NLP tasks into a single text-to-text format, enabling it to perform translation, summarization, and question answering tasks by converting the input text into a corresponding output text.

It has numerous points of strength. Firstly, T5 has a unified approach that simplifies the model architecture and training process, making it more efficient and scalable (Fedus et al., 2022). Secondly, the model has demonstrated competitive performance on multiple NLP benchmarks, such as the LAMBADA language modelling task and the SuperGLUE benchmark (Du et al., 2021). Thirdly, T5 is highly customizable, allowing developers to fine-tune the model for specific chatbot applications and requirements (Liu et al., 2021).

However, it also has some weak points. Firstly, like other LLMs, T5 demands considerable computational resources for training and deployment (Li, 2023). Secondly,

although the text-to-text format of T5 simplifies the training process, it may not be optimal for every NLP task or chatbot application (Lin et al., 2021). Thirdly, T5 may also exhibit biases present in its training data, which could impact its performance and response quality (Katsarou et al., 2022).

T5 is relatively less suitable for chatbot applications that require text generation, text understanding, and text manipulation, such as machine translation, summarization, question-answering systems, and content generation.

2.2.1.4 Robustly-optimized BERT approach (RoBERTa) by Facebook

RoBERTa, developed by Facebook AI, is an optimized version of BERT that incorporates various training and architectural improvements as well as training on a greater magnitude of data. RoBERTa is designed to capture context and relationships between words better, making it highly effective in sentiment analysis and named entity recognition tasks.

It has several areas of strength. Firstly, the training improvements of RoBERTa, including larger batch sizes and longer training times, contribute to its enhanced performance compared to the original BERT model (Liu et al., 2019). Secondly, the model has demonstrated state-of-the-art performance on various NLP benchmarks, such as GLUE and RACE (Liu et al., 2019). Thirdly, like BERT, RoBERTa can be fine-tuned for specific chatbot applications, providing developers with flexibility in customizing the model (Liu et al., 2019).

It also has many weaknesses. Firstly, RoBERTa, similar to BERT, is not inherently designed for text generation, which may limit its usefulness in generating responses for chatbot applications (Koroteev, 2021). Secondly, RoBERTa may also suffer from biases present in its training data, which could affect its performance and response quality (Silva et al., 2021).

Accordingly, RoBERTa is well-suited for chatbot applications that focus on understanding user input and extracting relevant information, such as question-answering systems, sentiment analysis, and information retrieval.

2.2.1.5 Sentence Transformer Models by Hugging Face

Sentence transformers, developed by Hugging Face, are a family of transformer-based models specifically designed for tasks that involve computing semantic sentence embeddings. These models are fine-tuned on tasks such as natural language inference and semantic textual similarity, enabling them to generate fixed-sized sentence embeddings. Conceptually, it is most similar to OpenAI's text embedding models, for instance, 'Ada'.

Its areas of strength are as follows. Firstly, sentence transformers generate fixed-sized embeddings, making it easier to compute sentence similarities and enabling their use in tasks such as semantic search and clustering (Zoupanos et al., 2022).

Secondly, these models can be fine-tuned on domain-specific data to create more accurate and relevant sentence embeddings for specific chatbot applications (Chandrasekaran & Mago, 2020).

It also has some inherent shortcomings. Firstly, sentence transformers are primarily designed for computing sentence embeddings and may not be as effective in text generation tasks as compared to models like GPT-3 or T5 but could complement other modules of LLM based conversational agents. Secondly, the model may also suffer from biases present in its training data, which could impact the quality of the generated embeddings (Nikolaev & Padó, 2023). Thirdly, while the Hugging Face ecosystem simplifies usage, models of Sentence Transformers may still require computational resources in case of independent deployment. However, HuggingFace provides the service of deploying the models in the ‘Spaces’ section of their web-based application (HuggingFace, 2023).

Sentence transformer models are ideal for chatbot applications that rely on semantic understanding, such as semantic search, clustering, and recommendation systems.

2.2.1.6 Summary of LLMs

Table 3 below presents the discussed summary of LLM models.

Model	Developer	Launch Date	Key Strengths	Preferred Use Cases
GPT Series	OpenAI	June 2020	Highly coherent responses, few-shot learning, hosted API, and multilingual	Customer support, virtual assistants, content generation, text embeddings
BERT	Google	October 2018	Bidirectional training, understanding input, open source	Sentiment analysis, question answering, information retrieval
T5	Google	October 2019	Unified text-to-text approach, versatility, open source	Translation, summarization, question-answering, content generation
RoBERTa	Facebook AI	July 2019	Training improvements, understanding input, open source	Sentiment analysis, question-answering, information retrieval
Sentence Transformers	Hugging Face	2019	Fixed-sized embeddings, semantic understanding, open source	Semantic search, clustering, recommendation systems

Table 3: A summary of the discussed LLM models for potential use in the chatbot implementation approach proposal.

This summary table presents an overview of the discussed LLM models, their developers, launch dates, key strengths, and preferred use cases. By considering these factors, developers can make informed decisions when selecting and implementing LLMs for chatbot applications, ultimately creating more effective and engaging conversational agents.

2.2.2 Selection of NLP Models for Use in Chatbots

The choice of an appropriate LLM for a chatbot application depends on various factors, such as the specific requirements of the application, the available computational resources, and the desired level of customization. On the one hand, the GPT series offers remarkable text generation capabilities and is suitable for applications that demand highly natural and contextually accurate responses. On the other hand, BERT and RoBERTa are more focused on understanding user input and are well-suited for tasks such as sentiment analysis and question answering. Similarly, the unified text-to-text approach of the T5 models makes it a versatile option for a wide range of chatbot applications that involve text generation, understanding, and manipulation.

When selecting an LLM for a chatbot application, developers should carefully consider the strengths and weaknesses of each model and the potential impact of exhibited biases in the training data. Furthermore, ongoing research and development of LLMs are likely to yield new models and improvements, which will further expand the possibilities for chatbot applications in the future.

Table 4 illustrates the comparative performance evaluation extracted from the GPT 4 technical report by OpenAI (2023c).

	GPT-4 Evaluated few-shot	GPT-3.5 Evaluated few-shot	LM SOTA Best external LM evaluated few-shot	SOTA Best external model (incl. benchmark- specific tuning)
MMLU Multiple-choice questions in 57 subjects (professional & academic)	86.4 % 5-shot	70.0 % 5-shot	70.7 % 5-shot U-PaLM	75.2 % 5-shot Flan-PaLM
HellaSwag Commonsense reasoning around everyday events	95.3 % 10-shot	85.5 % 10-shot	84.2 % LLaMA (validation set)	85.6 % ALUM
WinoGrande Commonsense reasoning around pronoun resolution	87.5 % 5-shot	81.6 % 5-shot	85.1 % 5-shot PaLM	85.1 % 5-shot PaLM

Table 4: Performance of LLMs on Academic Benchmarks as reported in the Technical Report by OpenAI (2023c) as of 14th March 2023. Comparison is between GPT-4, GPT-3.5, the best SOTA language models.

In conclusion, the comparison of existing LLM models for chatbot applications offers valuable insights into their relative strengths, weaknesses, and suitability for various use cases. By understanding these differences, developers can make informed decisions when selecting and implementing LLMs in their chatbot applications, ultimately creating more effective, efficient, and engaging conversational agents.

For the scope of the approach proposed in this thesis review, the latest GPT 4 and GPT 3.5 LLM models by OpenAI are considered due to their sufficiently abled performance in the evaluation as per the **Table 4**. Moreover, a number of risk mitigation techniques have been employed by OpenAI in their LLM models that enable them to be relatively less biased, less prone to hallucinations, and less suggestive of harm and harmful content. The technical paper by OpenAI (2023c) has detailed

quantitative evaluations exploring the risks mentioned. Another aspect taken into consideration was the easy access via API to the GPT model(s).

All the discussed reasons make them potentially plausible LLM technologies to power this approach proposal for the next generation of intelligent chatbots. Some of the methods explored for using GPT in chatbots include the domains of accounting (Alshurafat, 2023), psychology (Uludag, 2023), patient care (Hopkins et al., 2023), in library and information centers, language editing in scientific articles (Kim, 2023), in intelligent vehicles (Du et al., 2023), medicine (King, 2022), search engines (Microsoft, 2023), semantic database management (Lin et al., 2023), and orthopaedic research (Ollivier et al., 2023).

2.2.3 Domain-Specific Models

Domain-specific models are mostly built primarily upon the same principle architectures as analyzed in **Section 2.1.1**. However, due to specialized techniques employed in training the models to respond more specifically, the models are well suited per their language, field of application, and the task of application (Rahali & Akhloufi, 2023).

Rahali and Akhloufi (2023), in their review, discuss language-based models. First are the multilingual models, which carry the benefit of facilitating cross-linguistic transfer learning. Principally, despite the training on a task in one language, they can exhibit matchable performance in another language. An example is the multilingual variant of BERT, the mBERT, introduced by Devlin et al. (2018), where the same architecture of BERT was pre-trained on 104 monolingual corpora of different languages. Similarly, the multilingual T5 model, mT5, was trained on a dataset including 101 languages (Xue et al., 2020). Monolingual models are also used for language-specific tasks. These models are trained just on specific language corpora. BERT is an architecture used for monolingual models as well. Examples of monolingual language models include RomanianBERT (Dumitrescu et al., 2020), AraBERT (Antoun et al., 2020a), SweedishBERT (Malmsten et al., 2020), and FinnishBERT (Virtanen et al., 2019), among several others. A model, AraGPT2, a GPT-based model for Arabic NLG, was also introduced by Antoun et al. (2020b). It has a size of 77 GB and 8.8 billion words from the most extensive publicly available collection of filtered Arabic corpora that served as the model's training data.

In their paper, Rahali and Akhloufi (2023) also mention the models that are classified based on their field of application. Social media is one of those fields. In it, several tasks are automated based on AI classification techniques, such as sensitive language identification (Sawhney et al., 2022) (Thang Ta et al., 2022), recommendation tasks (Aljohani et al., 2022), and fake news detection tasks (Mehta et al., 2021) (Ghorbanpour et al., 2021). An example of a model in this domain is HateBERT – it identifies and classifies text based on the use of abusive English language (Caselli, Basile, Mitrović, & Granitzer, 2020). On hate speech datasets, such as OffensEval (Zampieri et al., 2019), AbusEval (Caselli, Basile, Mitrović, Granitzer et al., 2020), and HatEval (Basile et al., 2019), it consistently beats the standard BERT model at

recognizing offensive language. A RoBERTa-based model, RoBERTa-Twitter, was developed using 58 million tweets characterized and analyzed using the TweetEval benchmark (Barbieri et al., 2020). A similar strategy was executed for RoBERTa-Reviews and RoBERTa-News (Gururangan et al., 2020).

Another domain of application where LLMs are specifically used is computer programming. The approach used is similar in nature to training models on regular text. For this domain, a variety of base architectures are also used. The works of software code processing include CodeGPT (Lu et al., 2021), CodeBERT (Feng et al., 2020), and CoText (Phan et al., 2021).

Scientific research is a domain of application where models are trained for named entity recognition (extraction of information from text), scientific experiment prediction (Parrilla-Gutierrez, 2022), and information retrieval (Tian & Wang, 2021). Models such as SciBERT (Beltagy et al., 2019) and BioBERT (Lee et al., 2019) have performed well in scientific NER.

With the exploration of the domain-specific applications of specialized LLMs based on model architectures discussed in **Section 2.2.1**, it is noticeable that a model basis of GPT can be utilized for the approach of a chatbot. Accordingly, it also gives further possibilities of making it specifically tuned to respond in a more refined manner for precise domain applications, as noticed in the case of CodeGPT (Lu et al., 2021).

2.2.4 Limitations of LLMs

OpenAI's GPT-3, GPT -4 and other LLMS have made significant strides in NLP tasks and are widely utilized in various novel chatbot applications. However, despite their impressive capabilities, models like the GPT series have several limitations that developers and researchers must be aware of when considering their implementation in chatbot systems. This critical analysis of the case of GPT explores the limitations, implications, and potential future research directions of LLMs.

Firstly, **inconsistency and incorrect response generation** are significant limitations. Although GPT can generate coherent and contextually accurate responses, it occasionally produces inconsistent, incorrect, or nonsensical outputs. The primary objective of the GPT series during training is to minimize the perplexity of the generated text, which may result in responses that seem plausible but are factually incorrect or inconsistent with previous responses. A detailed consistency analysis is performed and reported by Jang and Lukasiewicz (2023), explaining this phenomenon in detail using experimental analysis. In a chatbot system, this may lead to user dissatisfaction and a decline in perceived reliability.

Secondly, **lack of control and explainability** is another limitation. GPT operates as a black-box model, making understanding and controlling the underlying mechanisms that generate its responses challenging. Consequently, it becomes difficult to ensure that the responses of GPT adhere to specific guidelines or follow a particular conversational strategy. Wahde and Virgolin (2021) outline principles for interpreta-

ble and safe conversational AI in their position paper. This aspect is severely highlighted as an adversary to safety-critical applications. This lack of control may cause problematic responses for chatbot systems in regulated industries or those that require strict compliance with established protocols and legal frameworks, such as the HIPPA laws in the US (U.S. Department of Health & Human Services: Office for Civil Rights, 2021).

Sensitivity to input phrasing is an inherent weakness. The text generation performance of GPT can be sensitive to slight changes in input phrasing, leading to varying response quality (Jang & Lukasiewicz, 2023). This sensitivity might result in different responses for semantically similar user inputs for chatbot applications, which could be confusing, misleading, or frustrating for users.

The **potential for biased and offensive content** is another associated risk. GPT models are trained on vast amounts of text data from the internet, which may include biased, offensive, or inappropriate content. As a result, the model may inadvertently generate responses that reflect these biases or contain offensive language. This limitation raises ethical concerns and poses challenges for implementing GPT-based models in chatbot systems that cater to diverse user groups and adhere to community guidelines. However, concrete improvements can be traced with the latest model releases by OpenAI to reduce this risk, as reported by OpenAI (2023c).

The **limited knowledge base** is also a prevalent issue. While GPT models by OpenAI can adapt to various domains through few-shot learning, its performance may not be optimal for highly specialized, custom, or narrow domains (Marr, 2023). Developers may need to fine-tune the model using domain-specific data to improve its performance in specialized chatbot applications. However, fine-tuning GPT can be computationally expensive and may require access to large amounts of domain-specific data. Alternatively, they can mask the input query by embedding chunks of contextually relevant data from the custom knowledge bases as outlined by a similar approach, ‘HyDE’ (Hypothetical Document Embeddings) presented by Gao et al. (2022).

OpenAI's GPT models have demonstrated remarkable capabilities in NLP tasks, making it an attractive option for chatbot applications. However, developers and researchers should be aware of the model's limitations: black-box design, response inconsistency, lack of control, sensitivity to input phrasing, the potential for biased content, and limited domain adaptability. These limitations may impact the user experience and effectiveness of chatbot systems powered by GPT-3.

Future research could focus on addressing these limitations using novel approaches to develop more efficient and controllable LLMs, enhancing their domain adaptability, and exploring techniques to mitigate biases in generated content. By addressing these challenges, researchers can pave the way for more reliable, efficient, and ethically sound chatbot systems. This approach encompasses consideration to avert some of the discussed limitations.

3 A Proposed Approach for the Next Generation of Chatbots

This section first explores the existing approaches to enhancing ‘intelligence’ in dialogue systems. Then fits those approaches to the process blocks of a chatbot application for an enhanced abstractive understanding of the next generation of intelligent chatbots.

3.1 A Review of Approaches to Chatbot Intelligence

This section reviews the elementary works related to chatbot development, focusing on prevalent approaches, merits, and demerits.

As discussed earlier in **Section 2.1.1**, the less intelligent or **rule-based** approach chatbots rely on a set of predefined rules, templates, and decision trees to handle user inputs (see **Figure 2**). While they are easy to implement and maintain for simple tasks, they are limited to predefined rules and unable to handle complex language nuances. They can also become quite complex to handle quickly with increased logic trees. An example of this approach is the ELIZA computer program – an attempt at enabling natural language communication between a machine and a human by Weizenbaum (1966).

Another approach is the **statistical methods** approach. ML algorithms and statistical models are implemented in it to analyze and generate text. Examples include Hidden Markov Models (HMMs) (Blunsom, 2004), Naive Bayes classifiers (Kelly & Johnson, 2021), and Support Vector Machines (SVMs) (Daumé III, 2004) (Chen et al., 2017). They are most useful in NLU tasks such as classification or pattern matching tasks such as intent recognition. They can learn patterns in data without explicit rules and are scalable to larger datasets.

Deep learning techniques, as discussed through the models in **Section 2.2**, such as RNNs, LSTM networks, and Transformer architectures, are at the forefront of NLP tasks of NLG and NLU. These are particularly serviceable in chatbot communications. This is because they can learn complex language patterns and generate natural-sounding text with better context understanding and history management. However, they have their demerits, some of which are discussed in **Section 2.4.4**.

A high-level procedural method is **transfer learning** which involves pre-training a neural network on a large corpus of text and then fine-tuning it on a specific task or domain (Mo et al., 2016). This allows chatbots to leverage extensive natural language training from one dataset and adapt it to a specific application with another smaller dataset. This improves performance with relatively smaller amounts of labeled data, faster training times, and reduced computational requirements post initial training.

For feedback incorporation, **reinforcement learning** (RL) algorithms can be used to train chatbots, allowing them to learn from trial and error and optimize their re-

sponses over time. In the context of dialogues, similar to the ones in chatbots, RL was introduced by Li et al. (2016). This has enabled dynamic adaptation to user preferences and conversation context with more coherent responses. Its implementation requires effective reward functions and human input in distinct models (Li et al., 2016).

The **hybrid approach** in chatbot systems combines different techniques to leverage the strengths of each method (Serban et al., 2017). For example, a chatbot might use rule-based logic for handling simple queries and deep learning models for more complex conversations. Moreover, deep learning NLP models can also incorporate RL for enhanced performance, for instance, the ‘Decision Transformer’ framework presented by Chen et al. (2021). This can help achieve better overall performance by leveraging multiple techniques and provide flexibility in adapting to different use cases and domains without extensive data-based fine-tuned training. Using these approaches, the sections ahead will explore their usability in the process blocks of an intelligent chatbot architecture.

3.2 Crucial Process Blocks

This section dives deep into the essential process blocks of a state-of-the-art chatbot with a particular focus on advanced NLP techniques.

3.2.1 Input Preprocessing

The first point of processing in a chatbot starts as the user inputs the query in a human language. This input needs to be processed to become ‘clean data’ that can be input to the LLM for NLU and appropriate NLG. Certain preprocessing functions at this stage include removing stop words, removing capitalization, formatting the input as a string, removing newline characters, removing line breaks, and input labeling (Suta et al., 2020).

For NLG models, for instance, the GPT series by OpenAI, the mentioned input preprocessing such that the input is a human language string is sufficient. A critical limitation of such models is the limited length of the input or context window (OpenAI, 2023e). The input window length for the basic GPT-4 and GPT-3 series is 8192 and 4096 tokens, respectively (OpenAI, 2023e). OpenAI has its own tokenization method where a string of input text is chunked into tokens for internal NLU and pricing estimation of the models. As per OpenAI (2023f) pricing documentation, 1 token is approximately equal to 4 characters or 0.75 of a word. Hence, input truncation or chunking methods are also applied to adjust the input string to the context window of the model.

NLU models, such as the retrieval-based and text-to-embedding models, require further input preprocessing (Suta et al., 2020). In it, the input string is converted to a word embedding, a vector representation of that string to compare with other vectors

in a vector space (Suta et al., 2020). This comparison extracts semantically most relevant data chunks from the vector store.

3.2.2 Prompt Engineering and Contextualization

This section discusses the techniques for capturing and leveraging LLM performance through prompt engineering – designing of input prompts to program a model in a certain way – and contextualization – external context and information awareness for the model. Simply put, natural language instructions input to LLMs are called prompts.

A classification framework presented by White et al. (2023) attempts to categorize prompt patterns into 5 segments, as given in **Table 5** (White et al., 2023).

Category	Prompt Pattern Examples
Input Semantics	Meta Language Creation
Output Customization	Output Automator Persona Visualization Generator Recipe Template
Error Identification	Fact Check List Reflection
Prompt Improvement	Question Refinement Alternative Approaches Cognitive Verifier Refusal Breaker
Interaction	Flipped Interaction Game Play Infinite Generation
Context Control	Context Manager

Table 5: Prompt Patterns and Categories (White et al., 2023)

The first prompt category in **Table 5** is the ‘**Input Semantics**’ category. It handles how an LLM interprets input data. The prompt translates it into a semantic pattern that the LLM can digest to generate an output. It includes meta language creation pattern which focuses on creating a customized language that the LLM can utilize (White et al., 2023). An example of this is a prompt that derives or outputs a structured query language (SQL) query from a natural language input prompt.

Then, the ‘**Output Customization**’ category is listed. The prompts instructing the LLM to generate content customized in a certain fashion fall in this segment. Examples given in **Table 5** include the ‘Persona’ (giving the LLM a persona or a role to play), ‘Visualization Generator’ (generation of prompts to be fed in other visualization models, such as ‘DALL-E’ (OpenAI, 2022), ‘Recipe’ (prompts for listing steps to

a possible end goal), and ‘Template’ (focusing the LLM to fill in a certain stated template) (White et al., 2023).

Following it is the ‘**Error Identification**’ pattern in **Table 5**. This prompt pattern is used to refine the output generated by the LLM and resolve any errors in it. The ‘Fact Check List’ pattern involves extracting facts from the response of the LLM and checking for their authenticity, while the ‘Reflection’ pattern involves prompting the LLM to introspect on its response for error mitigation (White et al., 2023).

The ‘**Prompt Improvement**’ category is next, where the quality of the generated response is managed. This is accomplished by ‘Question Refinement’, where the input prompt is reworded for a better question, ‘Alternative Approaches’, where the LLM suggests alternative methodologies of responding to the same query, ‘Cognitive Verifier’, where the LLM generates further questions for the user if the input prompt is insufficient for responding, or ‘Refusal Breaker’, where the LLM is itself told to rephrase the user query for re-querying in case the LLM is unable to respond (White et al., 2023).

The ‘**Interaction**’ category is focused on human-LLM interaction. The example patterns include ‘Flipped Interaction’. It is when the LLM is prompted to ask questions, similar to a role-reversal scenario (White et al., 2023). Another pattern is ‘Game Play’ in which the LLM responds with questions in a game form. This is similar to a mix of flipped interaction with a persona of a game (White et al., 2023). Lastly, it is the ‘Infinite Generation’ category. It is when the LLM is prompted to generate a response infinitely (White et al., 2023).

At last, it is the ‘**Context Control**’ category. This group of prompt patterns is focused on controlling the contextual information in the input query from which the LLM derives its responses (White et al., 2023).

The aspects discussed in the subsections ahead are highly correlated to the discussed prompt categories presented by White et al. (2023). Therefore, a logical understanding of the prompts is crucial for effective chatbot conversation design.

3.2.2.1 Behavior Priming

For an effective chat conversation design, the dialogue exchange must have a close affinity to how humans converse or chat. With regard to this, it is recommended to prime the chatbot response flow with a persona-patterned prompt that makes the LLM act or respond in a certain fashion. For example, suppose the chatbot is destined to assist visiting customers on an e-commerce store’s webpage. In that case, it is vital to prime the LLM, such as the GPT 4 model, by prompting ‘Act as a customer assistant’ or ‘You are a customer assistant’. This will make the LLM responses in the chatbot attain a nuanced resemblance to the fashion in which a human assistant responds.

An interesting experimental proof of concept for behavior priming an LLM is presented by Madhurima Vardhan et al. (2023). Their research paper fused behavior science and LLM using infusion techniques to prime an LLM-based conversation agent for a better health and fitness coaching experience.

3.2.2.2 Zero-Shot and Few-Shot Prompting

Zero-shot and few-shot prompting are two prompting techniques for response priming of LLMs. Zero-shot learning, a term first coined by Palatucci et al. (2009), is an ML training methodology where the ML model is trained to classify and label objects not used in its training phase. In the context of LLMs, specifically the GPT series, zero-shot is a prompt-based learning practice when the LLM is instructed once to output a certain way. Then it is ready to do so with any previously unexposed inputs. Moreover, it is evaluated through research by Wei et al. (2021) that, with fine-tuning (training on a large number of specific data points), language models can respond to zero-shot prompts aptly. Hence, Wei et al. (2021) titled their research paper ‘Fine-tuned Language Models Are Zero-Shot Learners’.

Few-shot learning is another ML training methodology where the ML model is trained on only a few training samples with labelled data. In the context of LLMs, specifically the GPT series, few-shot is a prompt-based learning practice when the LLM is instructed with a few example prompts and responses before it is ready to deal with unexposed inputs. The research by Brown et al. (2020), titled ‘Language Models are Few-Shot Learners’, demonstrated that the GPT series models, specifically GPT-3, are exceptionally responsive to few-shot demonstrations.

Generally, **Figure 4** by Zhou et al. (2022) gives conceptual clarity to the concepts of shot prompting.

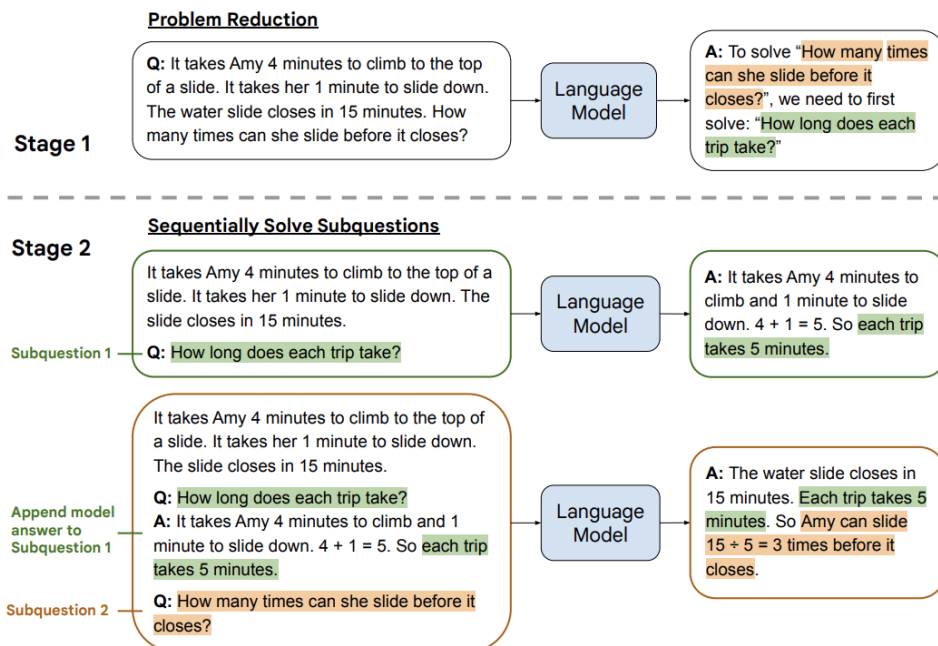


Figure 4: Prompt-based shot learning concept (Zhou et al., 2022)

In **Figure 4**, the stage 1 section shows an example of ‘least’ prompting, whereas stage 2 shows an example of ‘most’ prompting and their appropriate responses. Interestingly, the term ‘least-to-most prompting’ was adapted by Zhou et al. (2022) from educational psychology (Libby et al., 2008). In education psychology, this term refers to the technique of progressive instructions to enable a student to learn a new skill. The shot prompting used for LLMs can draw parallels from this human psychology technique.

In building conversation agents to respond in specific domain applications, the shot prompting technique is particularly useful in guiding the LLM NLGs to a particular pattern of text generation, similar to how a human child is taught. More pragmatic tutorials with a logical progression for prompting are also presented by Schulhoff et al. (2023).

3.2.2.3 Search & Retrieval from Connected Knowledge Bases

An essential block of a conversational agent for more valuable responses is its ability to incorporate external knowledge bases for response generation. This inherent limitation of LLMs is discussed in **Section 2.2.4**. One mechanism used to counter this limitation is fine-tuning the LLM with the appropriate custom dataset, as discussed in **Section 3.2.4**. This approach is mainly used in designing domain-specific LLMs, as exemplified in **Section 2.2.3**. Another solution to such a limitation is proposed by using a search and retrieval system to embed the results into the input prompt for the LLM. Ogundepo (2022) presents an exemplary architecture diagram in **Figure 5**.

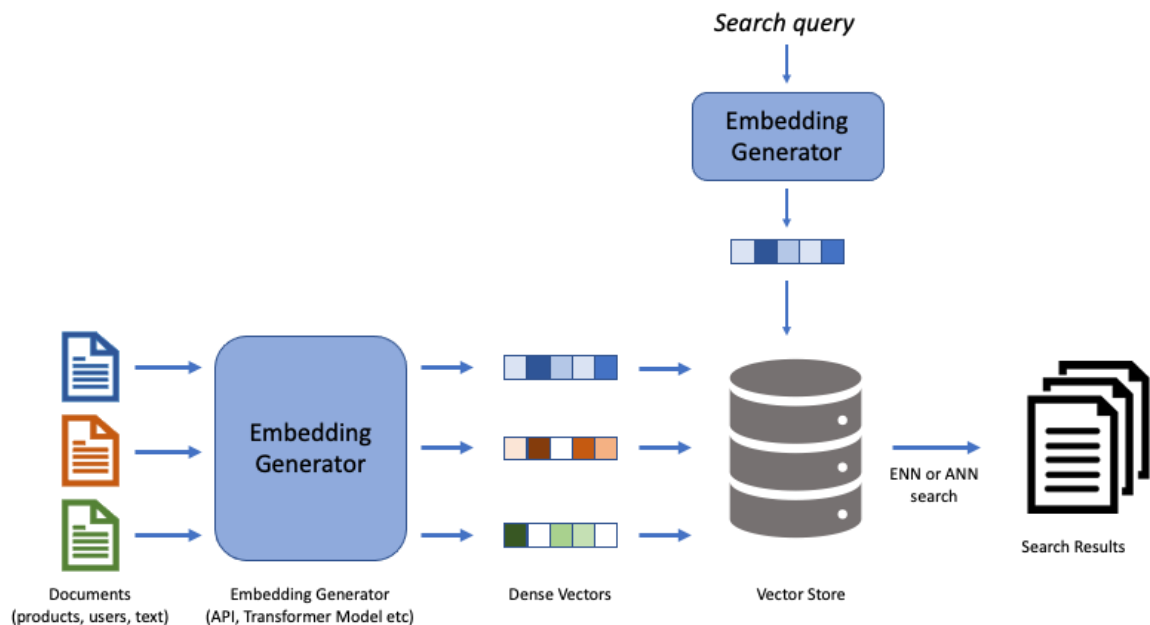


Figure 5: Architecture model for a semantic search and retrieval system (Ogundepo, 2022)

As explained in **Figure 5**, first, the documents from the external knowledge base are processed through a vector embedding generator model, such as the ‘text-embedding-ada-002’ model by OpenAI (2023a). The model returns dense vector lists

for the input documents. It should be noted that certain data preprocessing could be required to match the input specifications for the embedding generator LLM, such as chunking for a large corpus of text to meet the limited context window of the OpenAI embeddings model. After the creation of dense vectors, they are stored in a vector store – a database for storing, searching, and retrieving vectors. Current examples of vector stores particularly positioned for integration with LLM-based programs, such as chatbot apps, include Pinecone (Pinecone, 2023b), Chroma (Chroma, 2023), FAISS (Johnson et al., 2017), and Redis (Redis, 2023).

Once a user query is received, it is embedded using the same embedding generator to match dimensionality for the search. After the query’s vector is generated, it is sent to be matched against the vectors in the vector store. Hence, the metrics used to search the vector space include cosine similarity, Euclidean distance, and dot product similarity. **Table 6** explains these metrics for better understanding.

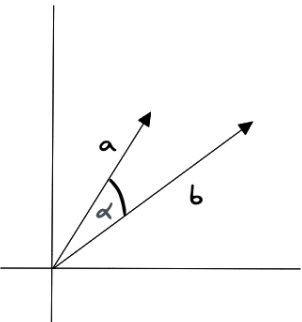
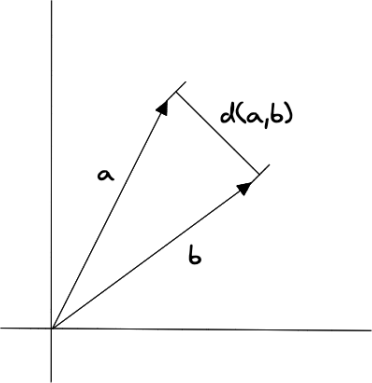
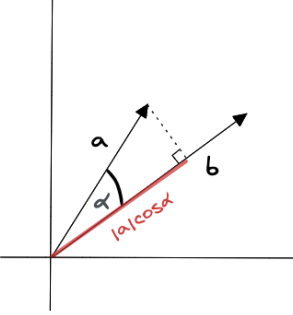
Vector Similarity Metric	Illustration in a 2-D Space	Explanation
Cosine Similarity		It is the angle measurement between two vectors in a multi-dimensional space. This metric only takes the angle into account. In the illustration it is signified by ‘ α ’.
Euclidean Distance		It is the straight-line distance between two vectors in a multidimensional space. This metric only takes the distance into account. In the illustration it is signified by ‘ $d(a,b)$ ’.
Dot Product Similarity		It is a metric that takes both, the angle and the distance, between two vectors into account. It is the product of the Euclidean magnitudes of the two vectors and the cosine of the angle in between. In the illustration it is signified by ‘ $ a \cos \alpha$ ’.

Table 6: Explanations of vector similarity metrics (Schwaber-Cohen & Pinecone, 2023)

The vector scores, based on the chosen metric score are then ranked, and a specified number of vector data with the highest metric score are retrieved to be sent to be entered in the prompt with the user query. This entire prompt is then sent to the LLM for appropriate NLU and NLG tasks. This is an approach of inputting the data of the external knowledge base to the LLM in the same prompt as the query is known as ‘In-Context Retrieval-Augmented Language Models’ Ram et al. (2023). Ram et al. (2023) have demonstrated improved LLM performance capabilities through their research using the mentioned method.

A crucial limitation of using the ranked vectors is the limited number of vectors to be accommodated in the limited context window of the LLM. Hence, careful size planning of data to embed is crucial for efficient process flow. Moreover, there can be a case where the vectors’ data size is more than the input context window can possibly accommodate. In that case, alternative approaches need to be developed. These could include setting a cutoff threshold of the metric for the accuracy or running LLM loops to extract relevant summarized information from all or a subset of the vectors. Conclusively, these strategies, approaches, and methods only devise so to include the knowledge of external knowledge basis into the LLM’s prompt for effective response generation by the LLM. The strategies for prompt injection-based knowledge base connection can be implemented architecturally, similar to the demonstration in **Figure 6**.

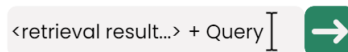


Figure 6: Retrieval text data injected into the LLM prompt

3.2.2.4 Maintaining Conversation History

Another critical aspect of a conversational agent is the ability to maintain chat context based on past messages sent in a chat. There are several methods through which conversational memory, or the context of the conversation, can be maintained in an LLM-powered chatbot application. A demonstrative example of the impact of conversational memory on LLMs response is given in **Figure 7**.

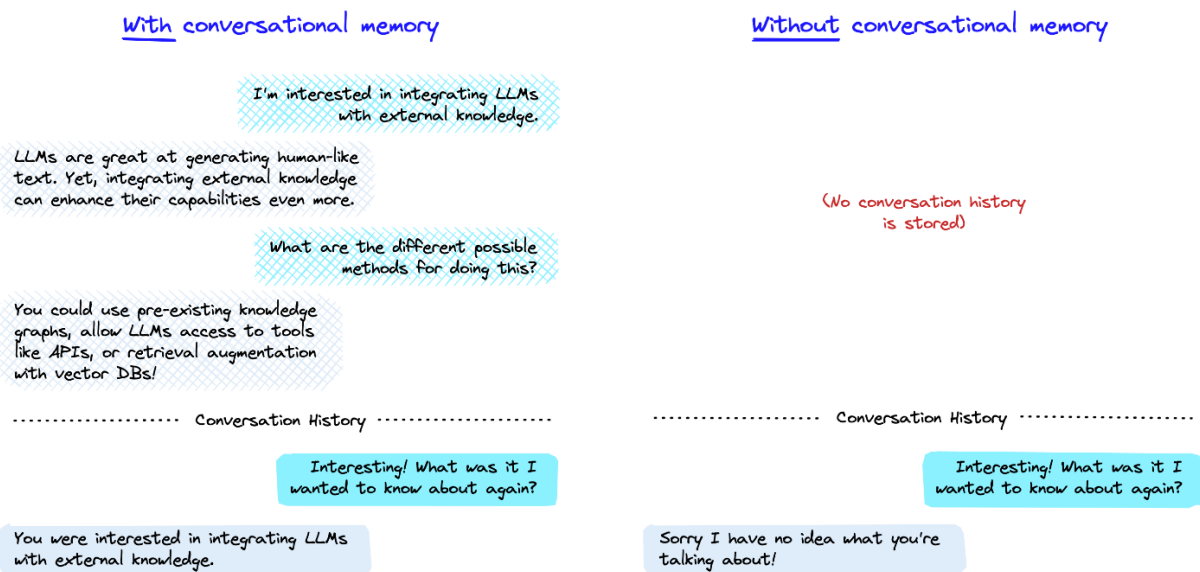


Figure 7: Effect on LLM response with and without conversational memory (Pinecone, 2023a)

Such an effect is important in a conversational agent to make it closer to how humans converse. The first method to implement such a conversational memory is injecting historical messages into the limited input prompt window (Pinecone, 2023a). However, this approach fails in the scenario of long conversations (Pinecone, 2023a). The second method involves injecting a limited number of past messages into the prompt window so that only a limited section of the limited input window is occupied by past conversations (Pinecone, 2023a). The third approach involves using a method involving prompt chaining (see **Section 3.2.3** for more on prompt chaining). All the conversation history is independently input into the LLM to generate a summary, and then that summarized chat history is injected into the prompt window (Pinecone, 2023a). However, this approach is inefficient with respect to relative response times due to the dual cycle run of the LLM (Pinecone, 2023a). The approaches described are the prevalent ways of powering LLM-based conversation agents at the time of this literature review. The prompt injection of chat history in the input window to the LLM is done similarly to the one shown in **Figure 8**.

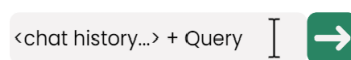


Figure 8: Chat history incorporation into the LLM prompt

3.2.3 Chain of Thought Prompting & Prompt Chaining

Chain of thought prompting is simply instructing an LLM to reach an end goal through a series of intermediate self-reasoning steps which has shown significant improvements in LLM's performance in complex tasks (Wei et al., 2022). Chain of thought prompting is often accomplished through a one-shot or a few-shot prompt-based learning examples. An exemplary chain of thought prompting effect is shown in **Figure 9**.

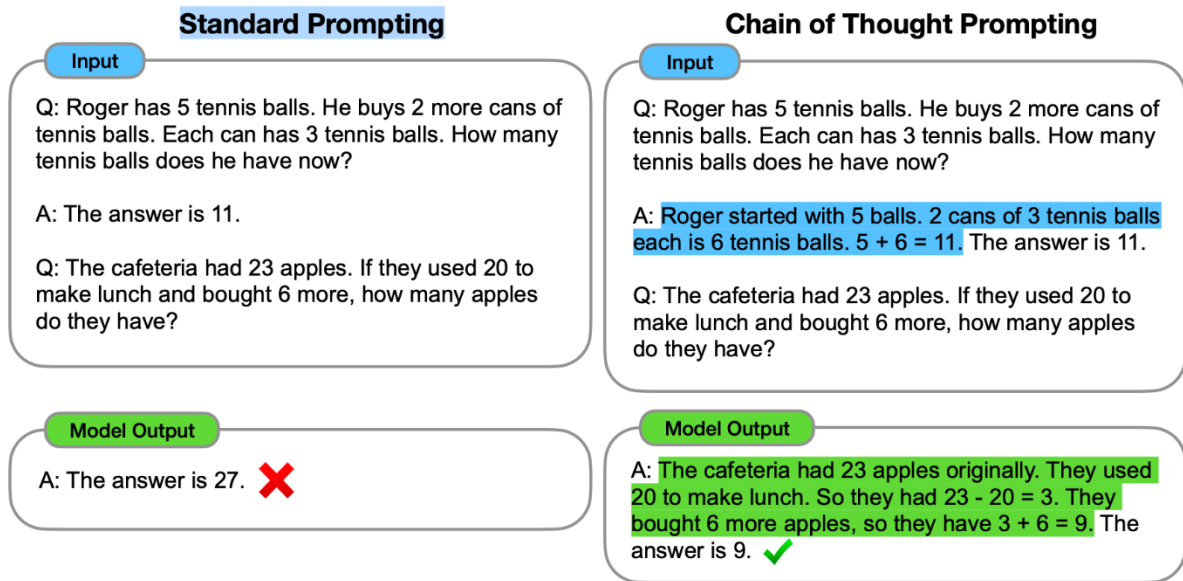


Figure 9: Chain of thought prompting effect using one shot prompt (Wei et al., 2022)

In **Figure 9**, the change in the depth of response reasoning using the chain of thought in the prompt is noticeable. It makes the LLM model reach the correct number of apples left. In chatbot applications, the chain of thought prompting can be imperative in applications requiring peculiar cognitive reasoning to reach a response.

Prompt chaining is a structural method used with LLMs to leverage LLMs in many real-world applications involving complex or multi-step tasks that are not otherwise possible through a single prompt (Wu, Jiang, et al., 2022). In other words, it is simply the linking of LLM runs together for complex multi-step processes where an output of one run of an LLM is an input to another run of an LLM using another or the same prompt (Wu, Jiang, et al., 2022).

In **Figure 10**, an example of a music chatbot application is presented by Wu, Jiang, et al. (2022), which is crafted using prompt chaining of LLM runs. It can be noticed in it that the user input (step 1) is first checked with another query to confirm whether it is related to music (step 2). Then if it is related to music, steps 3-5 or 8 are executed. However, if it is unrelated to music, it executes steps 9-10. Interestingly, the user input is also rechecked at step 10 by chaining the LLM runs to detect any ‘toxicity’ in the response of the user to respond appropriately.

An interesting step in this example is step 7, ‘JSYouTubeAPICall’. The information is extracted from the natural language user query to be sent as an API call to YouTube for appropriate data fetch. This technique of prompt chaining different tools via API access is a methodology for building intelligent agents capable of self-navigating and communicating with external APIs. It is one of the most promising features of modern LLMs that will lay the foundation for the next generation of intelligent chatbots and AGIs. According to Goertzel (2014), AGI’s can be characterized by capabilities that include reasoning, learning, communication, and actuation. Potentially, reasoning can be handled through recursive chaining, learning can be potentially handled

by RL, communication can be taken care of by NLG, and actuation can be implemented by interacting with external tools through automated API requests. Hence, powerful LLMs coupled with techniques, such as those discussed in this review, present an optimistic pathway of an AGI realization.

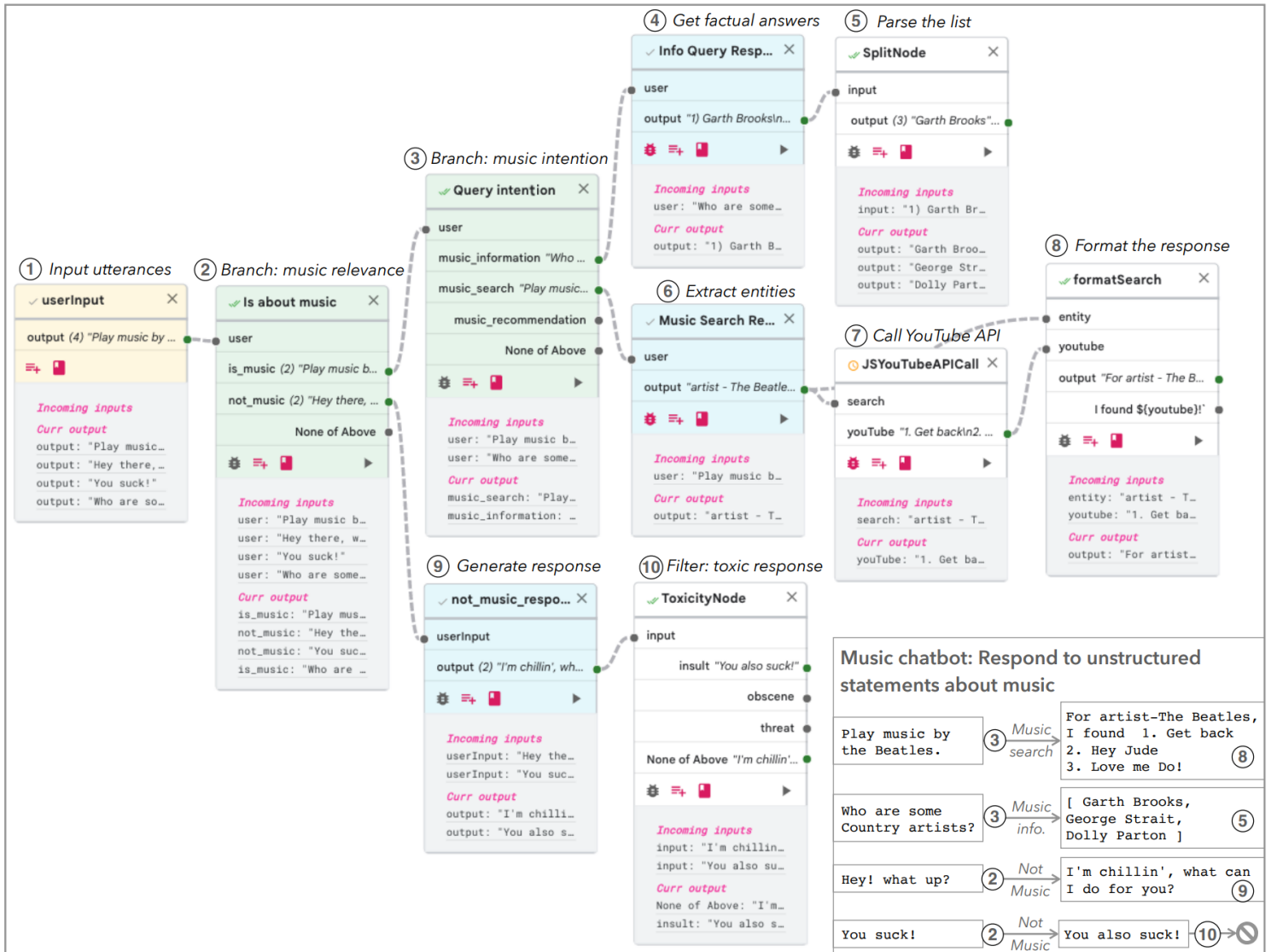


Figure 10: Prompt chaining example for a music chatbot (Wu et al., 2022)

3.2.4 Fine-Tuning

Fine-tuning LLMs with custom datasets has emerged as a popular technique to adapt pre-trained models to specific tasks or domains. LLMs have demonstrated impressive capabilities in a wide range of NLP tasks. However, out of the box, these models may not perform optimally for specific tasks or domains. Fine-tuning these models on custom datasets can help bridge the performance gap and tailor them to the desired use case. Fine-tuning involves training a pre-trained LLM on a custom dataset for a specific task or domain.

This process typically consists of the following steps. First is dataset preparation, in which a dataset relevant to the target task or domain is collected and preprocessed,

ensuring data quality and diversity. The second is appropriate model selection. Choosing an appropriate pre-trained LLM involves considering factors like model size, architecture, and pre-training data. Third is training and evaluation. Train the LLM on the custom dataset using techniques like transfer learning while monitoring performance metrics to ensure convergence and avoid overfitting.

Fine-tuning an LLM with custom datasets can have several impacts on model performance. Improved task-specific performance can be achieved. An example of fine-tuning an LLM model is demonstrated by Peng et al. (2023), LLaMA, which led to enhanced zero-shot learning capabilities using a dataset with 52000 English and Chinese data points. Fine-tuning can enhance the LLM's performance on the target task by adapting its knowledge to the specific domain, problem, or use case. The LLM is better able to generalize. Fine-tuning can improve the model's mapping of generalization on unseen data within the target domain or task. However, fine-tuning on small or biased datasets may result in overfitting, where the model performs poorly on unseen data. For example, to fine-tune the GPT series, it is recommended to use at least a few thousand data points in a dataset (OpenAI, 2023b), which can be achieved using the alternative techniques mentioned in **Section 3.2.2.3**. Another pain point can be maintaining and updating custom datasets, as it is labor-intensive and time-consuming. Hence, by carefully considering dataset quality, model selection, and fine-tuning strategies, practitioners can optimize the fine-tuning process and maximize the benefits the LLM can potentially provide. In turn, the quality of responses of the conversational agent will improve.

3.2.5 Countering Hallucinations & Inconsistencies

Hallucination is another aspect tied with LLMs, especially NLG tasks. Hallucinations refer to generating outputs that contain incorrect or fabricated information, while inconsistencies involve self-contradictory or contextually incoherent information in the generated text. Ji et al. (2022) conducted an extensive survey-based study to understand the types and scenarios of hallucinations. An example of hallucination was the input of 'Roger Federer' and 'Rafael Nadal' in the LLM, only to receive the output 'Roger Nadal' (Ji et al., 2022).

Delving deeper into these issues, reasons, and implications can help narrow towards potential solutions and improvements. Training data limitations are one of them. If the training data contains inaccuracies, biases, or noise, the model may learn to generate outputs that reflect these issues. Another aspect is the inability of LLMs to verify the factual aspect of the information generated. They rely on pattern matching and probability distributions, which can generate outputs that seem plausible according to the backend probability computations; however, they are incorrect or fabricated.

Hallucinations and inconsistencies can undermine user trust in the LLMs, leading to reduced adoption and limited utility in real-world applications. Moreover, when LLMs are put through decision-making processes, hallucinations and inconsistencies can result in incorrect or harmful decisions, potentially having severe consequences, especially in safety-critical scenarios. More importantly, generating inaccurate or

biased information raises ethical concerns, mainly when LLMs are used in sensitive domains, such as healthcare, finance, or politics.

With the awareness of such an inherent limitation of LLMs, strategies are researched to combat it. Ensuring the quality and diversity of the training data can help alleviate some of the issues related to hallucinations and inconsistencies. Practitioners can improve the model's performance by curating and preprocessing the data. As discussed in Section 3.2.4, fine-tuning LLMs on custom datasets or specific tasks can help them generate more accurate and contextually appropriate outputs. Researchers have explored various architectural modifications and techniques, such as memory-augmented networks or attention mechanisms, to mitigate hallucinations and inconsistencies in LLMs. Incorporating external knowledge sources like databases or knowledge graphs, as explained in Section 3.2.2.3, can help LLMs generate more accurate and relevant information. Introducing adversarial examples during the training process or prompt design to make the model more robust against generating hallucinations or inconsistencies through few-shot training or zero-shot training, as mentioned in Section 3.2.2.2, can also aid.

Moreover, popular LLMs, like the GPT series by OpenAI, allow experimenting with hyperparameters like temperature and top-k sampling to influence the randomness of the generated text. These can be adjusted to allow more focused responses that are less prone to hallucinations. Prompt chaining (see Section 3.2.4 for more on prompt chaining) can also be used to address hallucinations by the chain of thought prompting the LLM to question its own response for any fallacies. Lastly, human in the loop technique also exists to integrate human input into cases of incomplete knowledge for the LLM to complete response generation in a conversational agent.

3.2.6 RL and Feedback System

RL is a subfield of machine learning that focuses on training agents to make decisions by interacting with an environment. In RL, an LLM agent learns to perform actions to achieve a goal by maximizing the cumulative reward it receives from the environment. The learning process involves the agent exploring the environment, trying out different actions, and adjusting its strategy based on the rewards and penalties it receives.

The InstructGPT series were trained using RL techniques as outlined in the research by Ouyang et al. (2022). The effectiveness of the InstructGPT models is inculcated in them through RL. Many methods can be applied for the RL of LLMs, like policy-based methods or value-based methods. A much-used technique is also the RL with human feedback, where the LLM attempts to interact with the environment to maximize optimal output generation. However, alternative approaches need to be applied for black-box LLMs, such as the models in the GPT series. This is because the inability to change model parameters for RL is an innate feature of black box LLMs.

For black box LLMs, one of the techniques for feedback incorporation is laid out by Madaan et al. (2023), where an automatic feedback loop is run on the generated output by the LLM. An example of how this technique works is illustrated in **Figure 11**. On the input of 'Make the following sentence positive: *Pizza was bad*', it ran one

LLM cycle to generate the response ‘*Pizza was good*’. After that, the self-refine module was run to make the LLM self-analyze the response, generate feedback, and then automatically refine the output. After this iteration, the refined response ‘*Pizza was amazing*’ was generated.

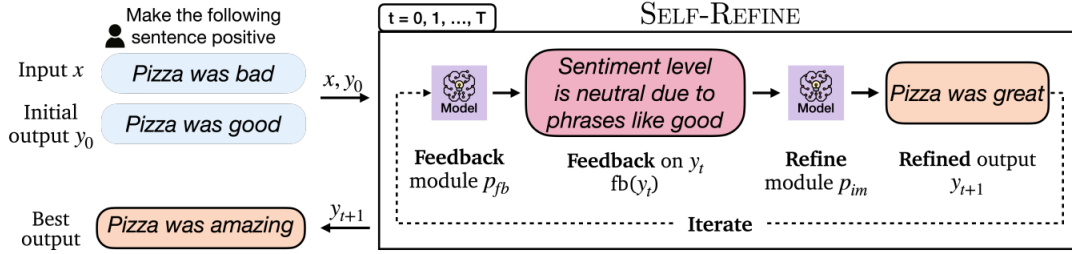


Figure 11: Self-refine algorithm of automatic feedback loop (Madaan et al., 2023)

An alternative to this approach can potentially be devised and tested on black box LLM models to augment them with feedback incorporation without requiring technical amendments like in conventional RL methods. Another feedback incorporation technique could be simply asking the user for feedback on the response and generating a refined response with that feedback. Moreover, building a vector store with similar model queries and responses could be made so that the LLM can generate a similar appropriate response. Therefore, a number of approaches can be formulated to incorporate feedback into the conversational agent and improve its responses.

3.3 Example Proposed Approach

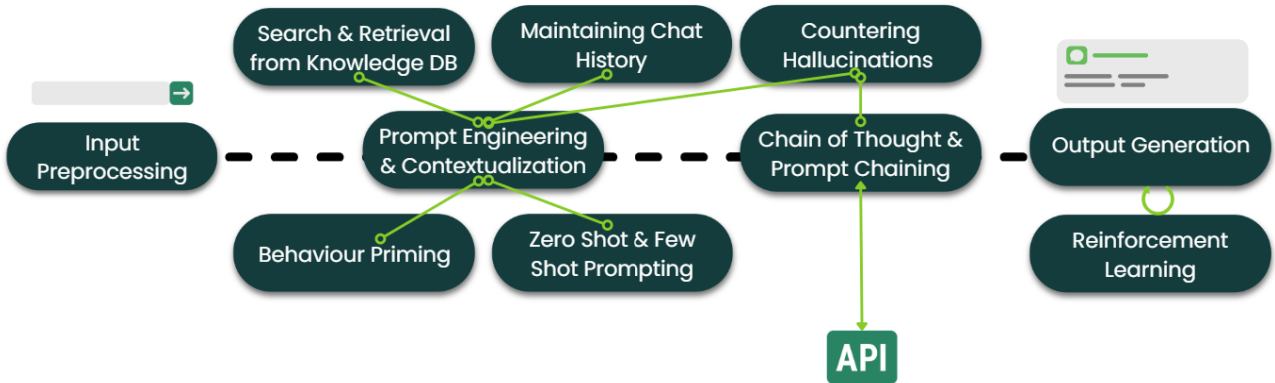


Figure 12: Concept map of the proposed LLM-based intelligent chatbot approach

Figure 12 gives an approach proposal for the next generation of intelligent chatbots and conversational agents that combine most of the modules introspected in **Section 3.2** into a singular approach. It starts with input preprocessing, and then aspects of prompt engineering and contextualization are incorporated to generate a response. Once generated, another chain could be linked for further response generation, optionally with external APIs. Before the final output generation, a hallucination-preventive mechanism could be applied for refinement. Once the output is generated, RL techniques and feedback incorporation mechanisms can be activated as per the reaction of the user to the response.

4 Discussion

With the advent of advanced LLMs, there is a plethora of promising research trajectories based on the fusion of AI with conversational agents. Improving prompt engineering methodologies, for instance, may entail investigating sophisticated contextualization approaches that allow chatbots to comprehend user inputs with greater precision to produce more accurate responses. Moreover, the optimization of fine-tuning procedures could concentrate on incorporating objectives tailored to specific tasks and employing active learning techniques, thereby enabling chatbots to better adapt to distinct conversational aims and user needs. In tackling the obstacles of hallucinations and inconsistencies in agent responses, researchers ought to investigate innovative strategies to alleviate these concerns, such as through code blocks similar to the concept of ‘guardrails’ (Burtenshaw, 2023) – that help validate and correct LLMs – or utilizing ‘multi-modal dynamic data sources’ – to furnish a more comprehensive context and enhance the dependability of the generated responses. An approach using prompt chaining has been utilized by Wu, Terry, et al. (2022) to make LLM usage more controllable and reliable, however, further research on more accurate methods is still required. This approach can be extended by recursive or looped chaining for complex actuation tasks.

The development of sophisticated RL techniques is essential for refining agent responses using real-world feedback, facilitating ongoing adaptation, and enhancement of LLMs. Establishing domain-specific models tailored to diverse industries and use cases may enable chatbots to attain greater efficacy within their designated domains. Research endeavors in this sphere should explore easing domain adaptation methodologies as well as devising automation framework for LLMs to interact with external API at self-disposal.

The ethical concerns associated with LLMs warrant thorough consideration, investigation, and preventive mechanisms. LLMs may inadvertently propagate biases present within their training data, leading to the generation of content that could perpetuate harmful stereotypes or exhibit preferential treatment towards specific demographics. Consequently, it is essential for researchers to develop robust methodologies for the identification, quantification, and mitigation of biases in LLMs, ensuring that the generated responses promote fairness and inclusivity. An example of this is the “Process for Adapting Language Models to Society (PALMS) with Values-Targeted Datasets” research paper (Solaiman et al., 2021) in which the OpenAI team explores techniques to the problem of inappropriate model outputs. Moreover, concerns related to privacy and data security arise, as LLMs may inadvertently learn from sensitive information present in the conversations, potentially exposing confidential or personally identifiable information. A recent case has been of Samsung employees using ChatGPT for code error removals inadvertently exposing chat data to OpenAI which it can use as training data for the model (Clarke, 2023). To address this issue, researchers should devise strategies for data control and anonymization built into the models architecture of deployment framework.

In addition, the application of LLMs in generating malicious content, such as disinformation, deepfakes, or socially engineered attacks, raises ethical concerns that demand the establishment of guidelines and regulations for responsible use. A recent popular movement has also been started by Elon Musk calling for AI development moratorium to constitute safe AI usage and development framework (Metz & Schmidt, 2023). Finally, the environmental impact of LLMs, stemming from the substantial use of such technologies. A research study exploring this area was conducted by Rillig et al. (2023), who have summarized it in **Figure 13**. The research (Rillig et al., 2023) explores the direct and indirect impact on environment as well as the impacts on environmental research. It is summarized in **Figure 13**. Prospective research could be directed at minimizing the adverse impacts.

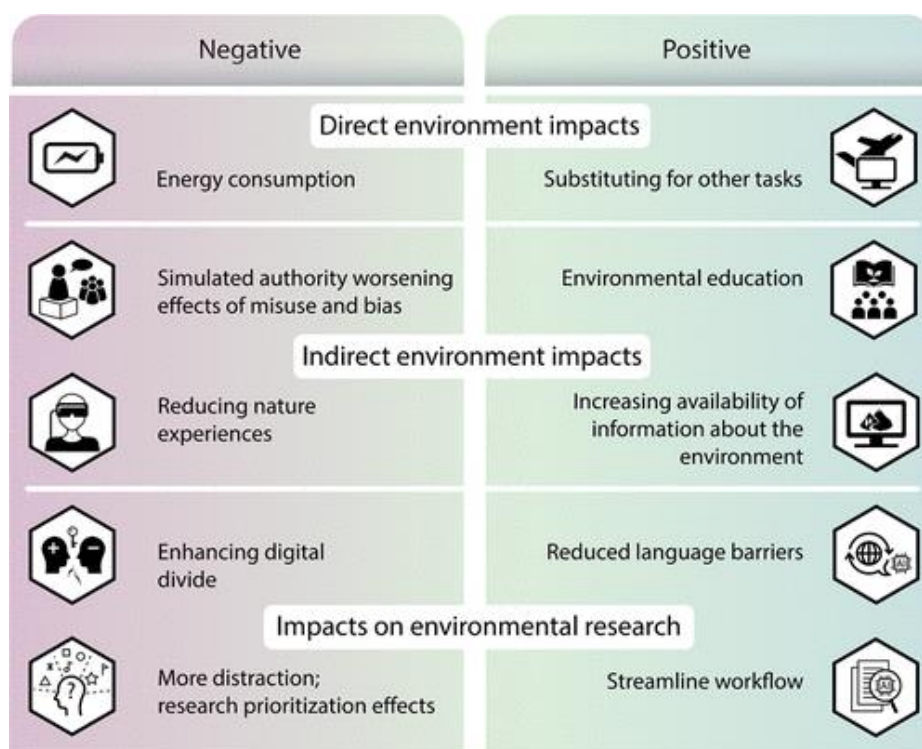


Figure 13: The risk and opportunities of LLMs with respect to the environment (Rillig et al., 2023)

By actively addressing the current limitations of LLMs and the subsequent ethical and environmental concerns, the AI research community can contribute to the responsible advancement, development, and deployment of LLMs. This aim of steering AI for better performance towards human intended goals is known as ‘AI alignment’ and is a research field (Yudkowsky, 2016). The better an AI system corresponds to human objectives, the better ‘aligned’ it is and vice versa. More aligned AI systems would ensure that their benefits are harnessed in an upright manner while minimizing potential adverse usage and subsequent consequences.

5 Conclusion

This SOTA literature review thesis explored the current state of research on the latest techniques for developing chatbot applications using NLP models. By examining scholarly works majorly from August 2016 to April 2023, the study identified techniques, methods, and limitations in the field and proposed a comprehensive approach to advance chatbot technology. The insights gained have the potential to help researchers, experts, and developers create the next generation of intelligent chatbots and advance the research endeavours for the development of AGI systems.

The thesis began with an introduction to chatbots and their underlying technologies, including an exploration of rules-based, pattern matching-based, and ML-based approaches. Building upon the ML-based approach, the focus was broadened to include generative LLMs and NLP models, namely GPT series, BERT, T5, RoBERTa, and Sentence Transformers. After discussing the selection of NLP models for chatbots and their limitations, **Section 2** was concluded. It then moved towards the building blocks of the next generation of chatbots in **Section 3**.

The building blocks of an intelligent conversational agent are comprised of crucial processes: input pre-processing, prompt engineering and contextualization, the chain of thought prompting, fine-tuning, countering hallucinations and inconsistencies, and an RL and feedback system. The literature review has also module-wise dissected the individual technologies involved and how they fit the context of a chatbot or conversational agent. The study then provided an illustrated example of a proposed approach to an intelligent conversational agent in **Figure 12**, which inculcated all the research points of the modules discussed earlier in a summarized form.

In conclusion, the literature review highlighted the advancements in chatbot and NLP technologies over the years, their limitations, and the potential path forward. A discussion section was also included highlighting the concerns and potential research directions concerning the thesis topic. The proposed approach represents a potential strategy to develop more advanced agents, which could significantly contribute to the evolution of prospective AGI systems. The insights and recommendations provided in this thesis can serve as a valuable resource for researchers and developers to bridge the gap between AI research and commercial usage.

Overall, the future of research on LLM-based agents is prosperous, with several avenues of groundbreaking advancements. By addressing these research areas and embracing a responsible and interdisciplinary approach, the development of the next generation of intelligent conversational agents has the potential to transform industries at a pace not realized before. It would be another step nearer to realizing AGI.

References

- Abid, A., Farooqi, M., & Zou, J. (2021, July 30). Persistent Anti-Muslim Bias in Large Language Models. *AIES '21: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. <https://dl.acm.org/doi/abs/10.1145/3461702.3462624>
- Acquire.io. (2022, January 17). *51 Amazing Chatbot Use Cases By Industry and Function*. Acquire.io; Acquire. <https://acquire.io/blog/chatbot-use-cases>
- Adamopoulou, E., & Moussiades, L. (2020a, May 29). An Overview of Chatbot Technology. *ResearchGate*. IFIP International Conference on Artificial Intelligence Applications and Innovations. https://www.researchgate.net/publication/341730184_An_Overview_of_Chatbot_Technology
- Adamopoulou, E., & Moussiades, L. (2020b). Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2, 100006. <https://doi.org/10.1016/j.mlwa.2020.100006>
- Aljohani, A., Rakrouki, M. A., Alharbe, N., & Alluhaibi, R. (2022). A Self-Attention Mask Learning-Based Recommendation System. *IEEE*, 10, 93017–93028. <https://doi.org/10.1109/access.2022.3202637>
- Alkaissi, H., & McFarlane, S. I. (2023). Artificial Hallucinations in ChatGPT: Implications in Scientific Writing. *Cureus*. <https://doi.org/10.7759/cureus.35179>
- Alshurafat, H. (2023). The Usefulness and Challenges of Chatbots for Accounting Professionals: Application On ChatGPT. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4345921>
- Antoun, W., Baly, F., & Hajj, H. (2020a). *AraBERT: Transformer-based Model for Arabic Language Understanding*. arXiv. <https://arxiv.org/abs/2003.00104>
- Antoun, W., Baly, F., & Hajj, H. (2020b). *AraGPT2: Pre-Trained Transformer for Arabic Language Generation*. arXiv. <https://arxiv.org/abs/2012.15520>
- Barbieri, F., Camacho-Collados, J., Neves, L., & Espinosa-Anke, L. (2020). *TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification*. arXiv. <https://arxiv.org/abs/2010.12421>
- Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., Rosso, P., & Sanguinetti, M. (2019). SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. *Proceedings of the 13th International Workshop on Semantic Evaluation*. <https://doi.org/10.18653/v1/s19-2007>
- Beltagy, I., Lo, K., & Cohan, A. (2019). *SciBERT: A Pretrained Language Model for Scientific Text*. arXiv. <https://arxiv.org/abs/1903.10676>
- Bhardwaj, R., Majumder, N., & Poria, S. (2021). Investigating Gender Bias in BERT. *Cognitive Computation*, 13(4), 1008–1018. <https://doi.org/10.1007/s12559-021-09881-2>
- Blunsom, P. (2004). Hidden Markov Models. In *Citeseer*. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=06fdod391377337a38e971456ef6b2d30652054>

- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., 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. M., Wu, J., Winter, C., & Hesse, C. (2020). *Language Models are Few-Shot Learners*. arXiv. <https://arxiv.org/abs/2005.14165>
- Burtenshaw, B. (2023a, March 20). *Detoxifying Large Language Models With Guardrails - Better Programming*. Medium; Better Programming. <https://betterprogramming.pub/detoxifying-large-language-models-with-guardrails-an-introduction-to-prompt-engineering-e20e1501de76>
- Burtenshaw, B. (2023b, March 20). *Detoxifying Large Language Models With Guardrails - Better Programming*. Medium; Better Programming. <https://betterprogramming.pub/detoxifying-large-language-models-with-guardrails-an-introduction-to-prompt-engineering-e20e1501de76>
- Caldarini, G., Jaf, S., & McGarry, K. (2022). A Literature Survey of Recent Advances in Chatbots. *Information*, 13(1), 41. <https://doi.org/10.3390/info13010041>
- Caselli, T., Basile, V., Mitrović, J., & Granitzer, M. (2020). *HateBERT: Retraining BERT for Abusive Language Detection in English*. arXiv. <https://arxiv.org/abs/2010.12472>
- Caselli, T., Basile, V., Mitrović, J., Granitzer, M., & Kartoziya, I. (2020). I Feel Offended, Don't Be Abusive! Implicit/Explicit Messages in Offensive and Abusive Language. *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 6193–6202. https://www.researchgate.net/publication/339799850_I_Feel_Offended_Don't_Be_Abusive_ImplicitExplicit_Messages_in_Offensive_and_Abusive_Language
- Chandrasekaran, D., & Mago, V. (2020). *Comparative analysis of word embeddings in assessing semantic similarity of complex sentences*. arXiv. <https://arxiv.org/abs/2010.12637>
- Chatfuel. (2023). *Chatfuel | Customer support and sales automation*. Chatfuel. <https://chatfuel.com/>
- Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., & Mordatch, I. (2021). Decision Transformer: Reinforcement Learning via Sequence Modeling. *Advances in Neural Information Processing Systems*, 34, 15084–15097. <https://proceedings.neurips.cc/paper/2021/hash/7f489f642a0ddb10272b5c31057f0663-Abstract.html>
- Chroma. (2023). *The AI-Native Open-Source Embedding Database*. Trychroma.com. <https://www.trychroma.com/>
- Clarke, P. (2023, April 4). *ChatGPT leaking Samsung chip secrets is iceberg's tip*. EeNews Europe. <https://www.eenewseurope.com/en/chatgpt-leaking-samsung-chip-secrets-is-icebergs-tip/>
- D'Avila, T. C. (2018). *KINO: an approach for rule-based chatbot development, monitoring and evaluation*. Repositorio Institucional Da UFMG. <https://www.semanticscholar.org/paper/KINO%3A-an-approach-for-rule-based-chatbot-monitoring-D%27C3%81vila/f44ab7517c37965dcoab9df46805fa2be2c86486>

- Dai, Z., & Callan, J. (2019). Deeper Text Understanding for IR with Contextual Neural Language Modeling. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 985–988.
<https://dl.acm.org/doi/abs/10.1145/3331184.3331303>
- Daumé III, H. (2004). Support Vector Machines for Natural Language Processing. In *Citeseer*.
<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=878b1c868799362ba09f962bb5b668dd140b82a0>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv.
<https://arxiv.org/abs/1810.04805>
- Du, H., Teng, S., Chen, H., Ma, J., Wang, X., Gou, C., Li, B., Ma, S., Miao, Q., Na, X., Ye, P., Zhang, H., Luo, G., & Wang, F.-Y. (2023). Chat with ChatGPT on Intelligent Vehicles: An IEEE TIV Perspective. *IEEE Transactions on Intelligent Vehicles*, 1–7.
<https://doi.org/10.1109/tiv.2023.3253281>
- Du, Z., Qian, Y., Liu, X., Ding, M., Qiu, J., Yang, Z., & Tang, J. (2021). All NLP Tasks Are Generation Tasks: A General Pretraining Framework. *ArXiv.org*. arXiv.
<https://arxiv.org/abs/2103.10360v1>
- Dumitrescu, S. D., Avram, A.-M., & Pyysalo, S. (2020). *The birth of Romanian BERT*. arXiv.
<https://arxiv.org/abs/2009.08712>
- Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *The Journal of Machine Learning Research*, 23(1), 5532–5270.
<https://dl.acm.org/doi/abs/10.5555/3586589.3586709>
- Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., & Zhou, M. (2020). *CodeBERT: A Pre-Trained Model for Programming and Natural Languages*. arXiv. <https://arxiv.org/abs/2002.08155>
- Gao, L., Ma, X., Lin, J., & Callan, J. (2022). *Precise Zero-Shot Dense Retrieval without Relevance Labels*. arXiv. <https://arxiv.org/abs/2212.10496>
- Ghorbanpour, F., Ramezani, M., Fazli, M. A., & Rabiee, Hamid R. (2021). *FNR: A Similarity and Transformer-Based Approach to Detect Multi-Modal Fake News in Social Media*. arXiv. <https://arxiv.org/abs/2112.01131>
- Gilliland, N. (2016, August 12). *Domino’s introduces “Dom the Pizza Bot” for Facebook Messenger*. Econsultancy; Econsultancy. <https://econsultancy.com/domino-s-introduces-dom-the-pizza-bot-for-facebook-messenger/>
- Goertzel, B. (2011, August 28). *Who coined the term “AGI”?* Goertzel.org.
<https://goertzel.org/who-coined-the-term-agi/>
- Goertzel, B. (2014). Artificial General Intelligence: Concept, State of the Art, and Future Prospects. *Journal of Artificial General Intelligence*, 5(1), 1–48.
<https://doi.org/10.2478/jagi-2014-0001>
- Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., & Smith, N. A. (2020). *Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks*. arXiv. <https://arxiv.org/abs/2004.10964>

- Hagendorff, T., Fabi, S., & Kosinski, M. (2022). *Machine intuition: Uncovering human-like intuitive decision-making in GPT-3.5*. arXiv. <https://arxiv.org/abs/2212.05206>
- Hopkins, A. M., Logan, J. M., Kichenadasse, G., & Sorich, M. J. (2023). Artificial intelligence chatbots will revolutionize how cancer patients access information: ChatGPT represents a paradigm-shift. *JNCI Cancer Spectrum*, 7(2). <https://doi.org/10.1093/jncics/pkado10>
- HuggingFace. (2022, November 7). *Sentence Transformers*. Huggingface.co. <https://huggingface.co/sentence-transformers>
- HuggingFace. (2023). *Spaces*. Huggingface.co. <https://huggingface.co/docs/hub/spaces>
- IBM. (2023). *IBM Watson*. Ibm.com. <https://www.ibm.com/watson>
- Jang, M., & Lukasiewicz, T. (2023). *Consistency Analysis of ChatGPT*. arXiv. <https://arxiv.org/abs/2303.06273>
- Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2022). Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12), 1–38. <https://doi.org/10.1145/3571730>
- Johnson, J., Douze, M., Jegou, H., & Meta. (2017, March 29). *Faiss: A library for efficient similarity search*. Engineering at Meta. <https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/>
- Katsarou, S., Rodríguez-Gálvez, B., & Shanahan, J. (2022). Measuring Gender Bias in Contextualized Embeddings. *AAAI Workshop on Artificial Intelligence with Biased or Scarce Data (AIBSD)*. <https://doi.org/10.3390/cmsf2022003003>
- Kelly, A., & Johnson, M. A. (2021). Investigating the Statistical Assumptions of Naïve Bayes Classifiers. *2021 55th Annual Conference on Information Sciences and Systems (CISS)*. <https://doi.org/10.1109/ciss50987.2021.9400215>
- Kim, S.-G. (2023). Using ChatGPT for language editing in scientific articles. *Maxillofacial Plastic and Reconstructive Surgery*, 45(1), 1–2. <https://doi.org/10.1186/s40902-023-00381-x>
- King, M. R. (2022). The Future of AI in Medicine: A Perspective from a Chatbot. *Annals of Biomedical Engineering*, 51(2), 291–295. <https://doi.org/10.1007/s10439-022-03121-w>
- Koroteev, M. V. (2021). *BERT: A Review of Applications in Natural Language Processing and Understanding*. arXiv. <https://arxiv.org/abs/2103.11943>
- Kucherbaev, P., Bozzon, A., & Houben, G.-J. (2018). Human-Aided Bots. *IEEE Internet Computing*, 22(6), 36–43. <https://doi.org/10.1109/mic.2018.252095348>
- Lai, G., Xie, Q., Liu, H., Yang, Y., & Hovy, E. (2017). *RACE: Large-scale ReAding Comprehension Dataset From Examinations*. arXiv. <https://arxiv.org/abs/1704.04683>
- Lee, A. (2023, January 26). *What Are Large Language Models Used For and Why Are They Important?* | NVIDIA Blog. NVIDIA Blog. <https://blogs.nvidia.com/blog/2023/01/26/what-are-large-language-models-used-for/>

- Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2019). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4), 1234–1240. <https://doi.org/10.1093/bioinformatics/btz682>
- Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., & Gao, J. (2016). Deep Reinforcement Learning for Dialogue Generation. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. <https://doi.org/10.18653/v1/d16-1127>
- Li, Y. (2023, February 28). *Business Analysis — AI Computational Cost*. Medium; Geek Culture. <https://medium.com/geekculture/business-analysis-ai-computational-cost-67a136957c95>
- Libby, M. E., Weiss, J. S., Bancroft, S., & Ahearn, W. H. (2008). A Comparison of Most-to-Least and Least-to-Most Prompting on the Acquisition of Solitary Play Skills. *Behavior Analysis in Practice*, 1(1), 37–43. <https://doi.org/10.1007/bf03391719>
- Lin, S., Hilton, J., & Evans, O. (2021). TruthfulQA: Measuring How Models Mimic Human Falsehoods. *ArXiv.org*. <https://arxiv.org/abs/2109.07958>
- Lin, W., Babyn, P., yan, Y., & Zhang, W. (2023). *Context-based Ontology Modelling for Database: Enabling ChatGPT for Semantic Database Management*. arXiv. <https://arxiv.org/abs/2303.07351>
- Liu, F., Huang, T., Lyu, S., Shakeri, S., Yu, H., & Li, J. (2021). EncT5: A Framework for Fine-tuning T5 as Non-autoregressive Models. *ArXiv.org*. arXiv. <https://arxiv.org/abs/2110.08426>
- 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. <https://arxiv.org/abs/1907.11692>
- Lu, S., Guo, D., Ren, S., Huang, J., Svyatkovskiy, A., Blanco, A., Clement, C., Drain, D., Jiang, D., Tang, D., Li, G., Zhou, L., Shou, L., Zhou, L., Tufano, M., Gong, M., Zhou, M., Duan, N., Sundaresan, N., & Deng, S. K. (2021). *CodeXGLUE: A Machine Learning Benchmark Dataset for Code Understanding and Generation*. arXiv. <https://arxiv.org/abs/2102.04664>
- Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegrefe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Welleck, S., Majumder, B. P., Gupta, S., Yazdanbakhsh, A., & Clark, P. (2023). *Self-Refine: Iterative Refinement with Self-Feedback*. arXiv. <https://arxiv.org/abs/2303.17651>
- Madhurima Vardhan, Hegde, N., Nathani, D., Rosenzweig, E., Karthikesalingam, A., & Seneviratne, M. (2023). Infusing behavior science into large language models for activity coaching. *MedRxiv*, 2023.03.31.23287995. <https://doi.org/10.1101/2023.03.31.23287995>
- Malmsten, M., Börjeson, L., & Haffenden, C. (2020). *Playing with Words at the National Library of Sweden -- Making a Swedish BERT*. arXiv. <https://arxiv.org/abs/2007.01658>
- Manychat. (2023). *Chat Marketing Made Easy with Manychat*. Manychat.com. <https://manychat.com/>

- Marr, B. (2023, March 6). The Top 10 Limitations Of ChatGPT. *Forbes*.
<https://www.forbes.com/sites/bernardmarr/2023/03/03/the-top-10-limitations-of-chatgpt/?sh=162a2ff98f35>
- Mehta, D., Dwivedi, A., Patra, A., & Anand Kumar, M. (2021). A transformer-based architecture for fake news classification. *Social Network Analysis and Mining*, 11(1).
<https://doi.org/10.1007/s13278-021-00738-y>
- Metz, C., & Schmidt, G. (2023, March 29). Elon Musk and Others Call for Pause on A.I., Citing “Profound Risks to Society.” *The New York Times*.
<https://www.nytimes.com/2023/03/29/technology/ai-artificial-intelligence-musk-risks.html>
- Microsoft. (2023). *Introducing the new Bing*. Bing.com. <https://www.bing.com/new>
- Mo, K., Li, S., Zhang, Y., Li, J., & Yang, Q. (2016). *Personalizing a Dialogue System with Transfer Reinforcement Learning*. arXiv. <https://arxiv.org/abs/1610.02891>
- Nikolaev, D., & Padó, S. (2023). *Representation biases in sentence transformers*. arXiv.
<https://arxiv.org/abs/2301.13039>
- Nimavat, K., & Champaneria, T. (2017). Chatbots: An overview. Types, Architecture, Tools and Future Possibilities. *ResearchGate*.
https://www.researchgate.net/publication/320307269_Chatbots_An_overview_Types_Architecture_Tools_and_Future_Possibilities
- Ogundepo, O. (2022, November 23). *Semantic Search With HuggingFace and Elasticsearch - Better Programming*. Medium; Better Programming.
<https://betterprogramming.pub/implementing-nearest-neighbour-search-with-elasticsearch-c59a8d33dd9d>
- Ollivier, M., Pareek, A., Dahmen, J., Kayaalp, M. E., Winkler, P. W., Hirschmann, M. T., & Karlsson, J. (2023). A deeper dive into ChatGPT: history, use and future perspectives for orthopaedic research. *Knee Surgery, Sports Traumatology, Arthroscopy*.
<https://doi.org/10.1007/s00167-023-07372-5>
- OpenAI. (2022). *DALL-E 2*. Openai.com. <https://openai.com/product/dall-e-2>
- OpenAI. (2023a). *Embeddings*. Openai.com.
<https://platform.openai.com/docs/guides/embeddings>
- OpenAI. (2023b). *Fine-tuning*. Openai.com.
<https://platform.openai.com/docs/guides/fine-tuning>
- OpenAI. (2023c). GPT-4 Technical Report. In *OpenAI*. <https://cdn.openai.com/papers/gpt-4.pdf>
- OpenAI. (2023d). *Introducing ChatGPT and Whisper APIs*. Openai.com.
<https://openai.com/blog/introducing-chatgpt-and-whisper-apis>
- OpenAI. (2023e, March 1). *OpenAI API*. Openai.com.
<https://platform.openai.com/docs/models/>
- OpenAI. (2023f, March 1). *OpenAI Pricing*. Openai.com. <https://openai.com/pricing>
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askeel, A., Welinder, P., Christiano, P. F., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 27730–27744.

- https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html
- Palatucci, M., Pomerleau, D., Hinton, G. E., & Mitchell, T. M. (2009). Zero-shot Learning with Semantic Output Codes. *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, 1410–1418.
<https://www.semanticscholar.org/paper/Zero-shot-Learning-with-Semantic-Output-Codes-Palatucci-Pomerleau/of6911bc1e6abee8bbf9dd3f8d54d40466429da7>
- Paperno, D., Kruszewski, G., Lazaridou, A., Pham, Q. N., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., & Fernández, R. (2016). *The LAMBADA dataset: Word prediction requiring a broad discourse context*. arXiv. <https://arxiv.org/abs/1606.06031>
- Parrilla-Gutierrez, J. M. (2022). Predicting Real-time Scientific Experiments Using Transformer models and Reinforcement Learning. *20th IEEE International Conference on Machine Learning and Applications (ICMLA)*.
<https://doi.org/10.1109/icmla52953.2021.00084>
- Peng, B., Li, C., He, P., Galley, M., & Gao, J. (2023). *Instruction Tuning with GPT-4*. arXiv. <https://arxiv.org/abs/2304.03277>
- Phan, L., Tran, H., Le, D., Nguyen, H., Anibal, J., Peltekian, A., & Ye, Y. (2021). *CoText: Multi-task Learning with Code-Text Transformer*. arXiv.
<https://arxiv.org/abs/2105.08645>
- Pinecone. (2023a). *Chatbot Memory with Langchain* | Pinecone. Pinecone.
<https://www.pinecone.io/learn/langchain-conversational-memory/>
- Pinecone. (2023b). *Vector Database for Vector Search* | Pinecone. Pinecone.
<https://www.pinecone.io/>
- Qasim, R., Bangyal, W. H., Alqarni, M. A., & Ali Almazroi, A. (2022). A Fine-Tuned BERT-Based Transfer Learning Approach for Text Classification. *Journal of Healthcare Engineering, 2022*, 1–17. <https://doi.org/10.1155/2022/3498123>
- Rahali, A., & Akhloufi, M. A. (2023). End-to-End Transformer-Based Models in Textual-Based NLP. *AI, 4*(1), 54–110. <https://doi.org/10.3390/ai4010004>
- Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. arXiv. <https://arxiv.org/abs/1606.05250>
- Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., & Shoham, Y. (2023). *In-Context Retrieval-Augmented Language Models*. arXiv.
<https://arxiv.org/abs/2302.00083>
- Ramesh, K., Ravishankaran, S., Joshi, A., & Chandrasekaran, K. (2017). A Survey of Design Techniques for Conversational Agents. *Communications in Computer and Information Science*, 336–350. https://doi.org/10.1007/978-981-10-6544-6_31
- Rasa Community. (2022, September 14). *Open Source Conversational AI*. Rasa; Rasa.
<https://rasa.community/>
- Redis. (2023). *Redis*. Redis. <https://redis.io/>
- Rillig, M. C., Ågerstrand, M., Bi, M., Gould, K. A., & Sauerland, U. (2023). Risks and Benefits of Large Language Models for the Environment. *Environmental Science & Technology, 57*(9), 3464–3466. <https://doi.org/10.1021/acs.est.3c01106>
- Sawhney, R., Neerkaje, A., & Gaur, M. (2022). A Risk-Averse Mechanism for Suicidality Assessment on Social Media. *Proceedings of the 60th Annual Meeting of the Associ-*

- ation for Computational Linguistics (Volume 2: Short Papers).
<https://doi.org/10.18653/v1/2022.acl-short.70>
- Schick, T., & Schütze, H. (2020). *It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners*. arXiv. <https://arxiv.org/abs/2009.07118>
- Schulhoff, S., Smith, W., Shikarpur, P., Hurst, J., Zhao, Z., Ahmed, W., & Si, C. (2023). *Learn Prompting*. Learnprompting.org. <https://learnprompting.org/>
- Schwaber-Cohen, R., & Pinecone. (2023). *Vector Similarity Explained*. Pinecone.
<https://www.pinecone.io/learn/vector-similarity/>
- Shaalán, K., Hassanein, A. E., & Tolba, F. (2017). Intelligent Natural Language Processing: Trends and Applications. In *Google Books*. Springer.
https://books.google.fi/books?id=eOI_DwAAQBAJ&dq=NLP+tasks+include+text+classification,+sentiment+analysis,+machine+translation,+speech+recognition,+information+extraction,+and+question-answering+systems+&lr=&source=gbp_navlinks_s
- Silva, A., Tambwekar, P., & Gombolay, M. (2021). Towards a Comprehensive Understanding and Accurate Evaluation of Societal Biases in Pre-Trained Transformers. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
<https://doi.org/10.18653/v1/2021.naacl-main.189>
- Singh, S., & Beniwal, H. (2022). A survey on near-human conversational agents. *Journal of King Saud University - Computer and Information Sciences*, 34(10), 8852–8866.
<https://doi.org/10.1016/j.jksuci.2021.10.013>
- Solaiman, I., Dennison, C., & OpenAI. (2021). *Process for Adapting Language Models to Society (PALMS) with Values-Targeted Datasets*. arXiv.
<https://arxiv.org/abs/2106.10328>
- Suta, P., Lan, X., Wu, B., Mongkolnam, P., & Chan, J. (2020). An Overview of Machine Learning in Chatbots. *International Journal of Mechanical Engineering and Robotics Research*, 9(4). <https://doi.org/10.18178/ijmerr.9.4.502-510>
- Thang Ta, H., Siddiquir Rahman, A. B., Najjar, L., & Gelbukh, A. (2022). Multi-Task Learning for Detection of Aggressive and Violent Incidents from Social Media. *CEUR Workshop Proceedings*, 3202. <https://ipn.elsevierpure.com/en/publications/multi-task-learning-for-detection-of-aggressive-and-violent-incid>
- Thorp, H. H. (2023, January 26). *ChatGPT is fun, but not an author*. Science.
<https://www.science.org/doi/full/10.1126/science.adg7879>
- Tian, X., & Wang, J. (2021). Retrieval of Scientific Documents Based on HFS and BERT. *IEEE*, 9, 8708–8717. <https://doi.org/10.1109/access.2021.3049391>
- Turing, A. M. (2007). Computing Machinery and Intelligence. *Parsing the Turing Test*, 23–65. https://doi.org/10.1007/978-1-4020-6710-5_3
- U.S. Department of Health & Human Services: Office for Civil Rights. (2021, June 9). *HIPAA Home*. HHS.gov. <https://www.hhs.gov/hipaa/index.html>
- Uludag, K. (2023). The Use of AI-Supported Chatbot in Psychology. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4331367>

- Van Dis, E. A. M., Bollen, J., Zuidema, W., Van Rooij, R., & Bockting, C. L. (2023). ChatGPT: five priorities for research. *Nature*, 614(7947), 224–226. <https://doi.org/10.1038/d41586-023-00288-7>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *Neural Information Processing Systems*, 30. https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
- Venkat, K. (2020, May 4). *Examples and Best use cases of chatbots in Banking Industry*. Medium; Voice Tech Podcast. <https://medium.com/voice-tech-podcast/examples-and-best-use-cases-of-chatbots-in-banking-industry-1a46cebdbocd>
- Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., Ginter, F., & Pyysalo, S. (2019). *Multilingual is not enough: BERT for Finnish*. arXiv. <https://arxiv.org/abs/1912.07076>
- Wahde, M., & Virgolin, M. (2021). The five Is: Key principles for interpretable and safe conversational AI. *2021 the 4th International Conference on Computational Intelligence and Intelligent Systems*. <https://doi.org/10.1145/3507623.3507632>
- Wallace, R. S. (2007). The Anatomy of A.L.I.C.E. *Parsing the Turing Test*, 181–210. https://doi.org/10.1007/978-1-4020-6710-5_13
- Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). *SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems*. arXiv. <https://arxiv.org/abs/1905.00537>
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*. arXiv. <https://arxiv.org/abs/1804.07461>
- Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., & Le, Q. V. (2021). *Finetuned Language Models Are Zero-Shot Learners*. arXiv. <https://arxiv.org/abs/2109.01652>
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. arXiv. <https://arxiv.org/abs/2201.11903>
- 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>
- West, D. M., & Allen, J. R. (2018, April 24). *How artificial intelligence is transforming the world*. Brookings; Brookings. <https://www.brookings.edu/research/how-artificial-intelligence-is-transforming-the-world/>
- 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. <https://arxiv.org/abs/2302.11382>
- Wu, T., Jiang, E., Donsbach, A., Gray, J., Molina, A., Terry, M., & Cai, C. J. (2022). *PromptChainer: Chaining Large Language Model Prompts through Visual Programming*. arXiv. <https://arxiv.org/abs/2203.06566>

- Wu, T., Terry, M., & Cai, C. J. (2022). *AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts*. arXiv. <https://doi.org/10.1145/3491102.3517582>
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., & Raffel, C. (2020). *mT5: A massively multilingual pre-trained text-to-text transformer*. arXiv. <https://arxiv.org/abs/2010.11934>
- Yudkowsky, E. (2016). *The AI Alignment Problem: Why It's Hard, and Where to Start*. <https://intelligence.org/files/AlignmentHardStart.pdf>
- Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). *SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffenseEval)*. arXiv. <https://arxiv.org/abs/1903.08983>
- Zhang, Z., Wu, Y., Zhao, H., Li, Z., Zhang, S., Zhou, X., & Zhou, X. (2020). Semantics-Aware BERT for Language Understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05), 9628–9635. <https://doi.org/10.1609/aaai.v34i05.6510>
- Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., & Chi, E. (2022). *Least-to-Most Prompting Enables Complex Reasoning in Large Language Models*. arXiv. <https://arxiv.org/abs/2205.10625>
- Zoupanos, S., Kolovos, S., Kanavos, A., Papadimitriou, O., & Maragoudakis, M. (2022). Efficient comparison of sentence embeddings. *Proceedings of the 12th Hellenic Conference on Artificial Intelligence*. <https://doi.org/10.1145/3549737.3549752>