



Enhancing text and voice based Chatbot(DBS specific) using Transformers and comparing it with traditional models LSTM

Sandeep Kumar

Applied Research Project submitted in partial fulfilment of the
requirements for the degree of

Master of Science in Data Analytics

at Dublin Business School

Supervisor: Dr Paul McEvoy

August 2025

Declaration

I, Sandeep Kumar, declare that this Applied Research Project that I have submitted to Dublin Business School for the award of Master of Science in Data Analytics is the result of my own investigations, except where otherwise stated, where it is clearly acknowledged by references. Furthermore, this work has not been submitted for any other degree.

Signed: Sandeep Kumar

Student Number: 20049275

Date: 25th August 2025

Acknowledgments

I would like to express my thanks to all those who supported me throughout the course of this Applied Research Project.

First and foremost, I would like to thank my project supervisor, Dr Paul McEvoy, for the guidance, support, and incredibly valuable feedback at every stage of this research. It has been a privilege and pleasure to work under his guidance. His insights have greatly contributed to shaping the quality of this work.

I would also like to acknowledge the faculty and staff at Dublin Business School for providing the academic environment, resources and knowledge that enabled me to successfully complete this project.

Finally, I owe my deepest gratitude to my family and friends for their support and constant encouragement throughout this study.

Contents

Contents	3
Abstract	5
1.0 Introduction	6
<i>1.1 Research Question</i>	8
<i>1.2 Research Objectives</i>	8
2.0 Literature Review	9
3.0 Research Methodology	16
<i>3.1 Research Philosophy</i>	17
<i>3.2 Research Approach</i>	19
<i>3.3 Research Design</i>	21
<i>3.4 Research Method</i>	22
4.0 Ethical considerations	24
5.0 Data Collection and Analysis	26
<i>5.1 Data Cleaning and Preprocessing</i>	27
<i>5.2 Exploratory Data Analysis (EDA)</i>	30
<i>5.3 Model Development</i>	31
<i>5.3 System Design and Implementation</i>	33
<i>5.4 Model Evaluation and Benchmark Comparison</i>	34
6.0 Conclusion	43
7.0 Limitations and Future Scope	44
8.0 Appendix A Sample Implementation and Examples	45
9.0 Appendix B Artefacts and Repositories details	49
11.0 References	53

List of Abbreviations

DBS.....	Dublin Business School
AI.....	Artificial Intelligence
IT.....	Information Technology
DBS.....	Dublin Business School
LSTM.....	Long Short-Term Memory
BLEU.....	Bilingual Evaluation understudy
BERT.....	Bidirectional Encoder representation from Transformers
RNN.....	Recurrent Neural Networks
GPT.....	Generative Pre-trained Transformer
AIML.....	Artificial Intelligence Markup Language
ALICE.....	Artificial Linguistic Internet Computer
XML.....	eXtensible Markup Language
NLP.....	Natural language processing
ML.....	Machine learning
TF-IDF	Term Frequency–Inverse Document Frequency
BoW.....	Bag-of-Words
GloVe.....	Global Vectors for Word Representation
LSA.....	Latent Semantic Analysis
GRU.....	Gated Recurrent Unit
NLTK	Natural Language Toolkit
EDA.....	Exploratory Data Analysis
FAISS.....	Facebook AI Similarity Search

Abstract

This work looks at the development of a chatbot tailored for Dublin Business School(DBS), using transformer-based, state-of-the-art architectures in language processing that render contextually relevant text (Behl and Bibhu, 2024) and compares their efficiency against “Long Short-Term Memory”(LSTM) network, architecture capable of capturing long range dependencies crucial for understanding context (Hochreiter & Schmidhuber, 1997). The methodology adopted in this study is quantitative in nature and aims to evaluate metrics such as response accuracy, inference times, Bilingual Evaluation understudy (BLEU) and Bidirectional Encoder representation from Transformers scores (BERTScore) , in order to answer research questions and objectives. Results indicate that the Transformer-based chatbot achieved higher accuracy and semantic scores than the LSTM model, though with tradeoff in inference time. The findings align with the established theory of transformers architecture outperforming recurrent models such as LSTM (Vaswani et al., 2017). However, limited scope of this work does not allow to incorporate qualitative aspects such as user experience, emotions or subjective feedback during interactions with chatbot. It is recognized that these dimensions would bring a more rounded aspect to the work, and further study would need to be undertaken to further this area. However, this study may provide easy, scalable and adaptable solutions for other institutions beyond DBS, offering a framework for broader educational chatbot implementations.

1.0 Introduction

AI assistants are a technology that has transformed the way businesses interact with their customers by providing self-service power to the users, helping them get what they want, when they want it, the way they want it (**Freed, 2021**). These assistants are saving significant labor and operational costs for businesses by automating routine enquiries and support tasks (**Patil et al., 2024**). One prominent example of AI assistants is Conversational AI, or chatbots, which fall under this category and use full conversational dialogue to accomplish one or more tasks (**Freed, 2021**).

Beyond business environments, the application of AI-powered chatbots has also seen rapid growth in the education sector in recent years particularly for providing student support and improving communication (**Tapia-Hoyos, 2021**). This growth is partly due to the fact that traditional communication between prospective students and universities is often handled manually, which can be a time-consuming process and a burden on admissions staff (**Nguyen, Le, Hoang, and Nguyen, 2021**).

These virtual assistants not only answer and manage queries but also offer a spectrum of services like academic tutoring, counselling, career guidance, automate routine tasks like ticket creation for administration and IT, access to academic records, or academic advising (**Mashilo et al., 2024**). This growing reliance stems from the need for 24/7 availability, scalability, and delivery of context-aware, consistent, and accurate responses—capabilities often lacking in traditional systems (**Patil et al., 2024**).

To appreciate how conversational AI reached its current capabilities, particularly in education, it is important to examine its evolution over time. The evolution of conversational AI span several decades (**Cambria and White, 2014**). Early systems, inspired by the Turing test

(Turing, 1950), relied on rule-based, scripted responses, as seen in pioneers like ELIZA (Weizenbaum, 1966) and PARRY (Colby, K., 1975). These systems could simulate conversation but were domain-specific and unable to maintain dialogue over extended interactions. They were rule based and had no contextual intelligence(Shum, He, and Li, 2018). Similarly, vector-based models such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) improved semantic representation but had limited ability to process multi-turn conversations effectively(Young et al., 2018). More recent advances in deep learning architecture, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, allowed better understanding of sequences and conversational flow, however, these models often suffer from vanishing gradients and are computationally inefficient for longer dependencies(Hochreiter & Schmidhuber, 1997).

These limitations in earlier models prompted researchers to explore more advanced architectures, most notably the Transformer-based model introduced by (Vaswani et al., 2017) in their seminal paper “Attention is All You Need.” In this paper they explained how the Transformer architecture uses self-attention mechanisms that allow the model to weigh the relevance of each word in a sentence relative to the others, thereby improving both processing efficiency and the preservation of conversational context that addressed many of the challenges by introducing parallel processing through self-attention, contextual coherence, and transfer learning(Behl and Bibhu, 2024). This progression underscores the significance of using a Transformer-based approach in enhancing chatbot capabilities within a domain-specific academic context like DBS, where student queries responses are grounded in institutional data.

Building on these technological advancements, the present research attempts to apply transformer-based architecture within academic domain, aiming to demonstrate its potential

advantages over traditional LSTM models(Behl and Bibhu, 2024) in developing a DBS-specific chatbot. Two chatbot variants, one using Transformers and the other using LSTM, are expected to be developed and evaluated based on quantitative performance metrics, including response exact accuracy, inference time, semantic scores, using a customized dataset derived from DBS websites, brochures, and institutional documentation.

Building on the discussion so far and considering the overall intention of this work, the following research questions and commensurate objectives are proposed.

1.1 Research Question

How can transformer-based architecture improve efficiency ("Attention is All You Need" by Vaswani et al. in 2017) and accuracy of a DBS-specific chatbot compared to traditional models like LSTM?

1.2 Research Objectives

1. Design and develop a DBS-specific chatbot using transformer-based and LSTM based models – This will allow implementation and attempt demonstration of comparable chatbot versions leveraging Transformers and LSTM models, respectively for evaluating their capabilities in responding to the student specific queries.
2. Compare the chatbot's performance against traditional models like LSTM(inference time , semantic scores, accuracy) – This comparative analysis will attempt to provide metrics essential for comparison between the chatbot versions, addressing directly the research question.

To build upon the research questions and objectives outlined above this literature review aims to explore the evolution of Conversational AI as a foundation for developing a DBS-specific chatbot using Transformer-based and LSTM models. To understand how the accuracy of the two

chatbot architectures can be compared, central to this study's research question, it is essential to trace the origins and progression of conversational systems. By examining the theoretical roots, design principles, benefits and limitations of early chatbots, we can delve deep into more advanced deep learning techniques and appreciate the motivations behind this transition.

2.0 Literature Review

The conceptual roots of the conversational systems trace back to the release of the paper "Computing Machinery and Intelligence" (Turing, 1950), where it was proposed, the now-famous Turing Test as a means to evaluate machine intelligence(Shum, He, and Li, 2018). In this he addresses and reframed the most fundamental question of artificial intelligence: can machine think? Into an imitation game which determines whether a machine can exhibit human behavior or not. He argued that if computers are given enough memory, machines can learn and simulate human reasoning (Turing, 1950). This laid the foundation for future AI systems, including chatbots (Saygin, Cicekli, & Akman, 2000).

Building on these ideas, the 1960s saw the creation of Eliza developed by Joseph Weizenbaum. In his paper it is explained that the keywords appearing in the input triggers the decompositions rules which help in analyzing the input sentence(Weizenbaum, 1966). Responses are then generated by reassembly rules associated with selected decomposition rules. For instance, if a user said, "It seems that you hate me," ELIZA could apply decomposition rules and reassemble a response like "What makes you think I hate you?" (Weizenbaum, 1966). In the context of this research, ELIZA highlights the limitations of early rule-based systems, particularly their inability to retain context or infer meaning, limited scope of knowledge(Shum, He, and Li, 2018), limitations which contemporary models like Transformers are designed to overcome through mechanisms like attention and contextual embeddings (Vaswani et al., 2017).

Advancing this theory, PARRY, developed by Kenneth Colby in 1975, attempted to simulate the conversational behavior of a person with paranoid schizophrenia (overly suspicious, anxious, or fearful person) (Shum, He, and Li, 2018). Unlike ELIZA, PARRY included a psychological model representing beliefs, fears, and emotional states, enabling it to generate responses not just based on surface input, but also on its internal affective state, allowing it to simulate context-dependent and emotionally reactive behavior (Colby, 1975). While Parry is still rule based, it makes a crucial step toward emotionally aware dialog systems (Shum, He, and Li, 2018). For this study, its significance lies in the shift from static to state-dependent behavior which is an important precursor to attention mechanisms found in modern Transformer models such as BERT and GPT (Vaswani et al., 2017; Jurafsky & Martin, 2025).

Another important advancement was the introduction of Artificial Intelligence Markup Language (AIML) by Richard Wallace and implemented in Artificial Linguistic Internet Computer (ALICE) entity. It provided a simplified XML-based framework for developing conversational agents, defining reusable objects such as categories and topics, enabling bots to recursively invoke pattern matchers (Wallace, 2003). AIML, however, like its predecessors, lacked any deep learning or self-learning capability and ALICE specifically failed Turing Test as it couldn't maintain a dialog for long period of time (Shum, He, and Li, 2018).

Taken together, these early innovations laid the groundwork for today's data-driven conversational systems, however, their limitations in handling long-term dependencies, and limited contextual information, necessitated a shift toward more robust architectures (Shum, He, & Li, 2018; Jurafsky & Martin, 2025; Young et al., 2018) particularly those powered by deep learning, such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks which process sequential data by remembering or maintaining memory over previous

tokens(Hochreiter & Schmidhuber, 1997; Cho et al., 2014) and, ultimately, Transformer-based models which increased the computation through parallel processing and could maintain long range dependencies (Vaswani et al., 2017). These transitions will be explored in the following sections, connecting the technological evolution of chatbots to the research aim of comparing Transformer and LSTM architectures in a DBS-specific context.

As conversational systems evolved beyond rule-based frameworks like ELIZA (Weizenbaum, 1966) and PARRY (Colby, 1975), a significant transformation occurred with the integration of machine learning and statistical natural language processing (NLP) techniques (Al-Amin et al., 2024;Cambria and White, 2014). In the early 2000s, researchers began shifting from static pattern-matching rules to data-driven approaches that could learn from real conversational data, and it marked the beginning of probabilistic intent recognition, where statistical models were employed to map user utterances to appropriate intents and responses, enhancing flexibility and performance (Jurafsky & Martin, 2025). This transition is particularly relevant to this research, which aims to compare the effectiveness of Transformer-based and LSTM-based models in a DBS-specific chatbot, as both architectures emerged from and build upon these NLP advancements.

The foundational structure for processing natural language is typically organized as pipeline which includes tasks such as text cleaning (stop word removal, punctuation filtering, and digit exclusion), sentence segmentation and word tokenization(breaking the text into smaller linguistic units) followed by stemming and lemmatization techniques(to reduce words to their base forms) (Elov, Khamroeva, and Xusainova, 2023). Understanding these stages is critical for constructing effective conversational agents and evaluating model architectures, as preprocessing directly influences downstream performance in both LSTM and Transformer-based systems.

The next critical stage in the NLP pipeline involves transforming preprocessed tokens into numerical representations suitable for machine learning models (Elov, Khamroeva, and Xusainova, 2023). Earlier methods like Bag-of-Words (BoW), represents word frequency in a document but fail to capture word order or semantics while another method TF-IDF (Term Frequency–Inverse Document Frequency) improved on this by measuring a word's importance relative to a document and corpus however, both BoW and TF-IDF generate high-dimensional and sparse representations (as the length of sentence increases the dimensions of the vectors also increases) (Manning, Raghavan and Schütze, 2008).

To overcome these limitations, the research moved toward word embeddings dense vector representations that capture both semantic and syntactic relationships that played a vital role in building continuing word vectors based on their context in large corpuses (Liu et al., 2015). Early techniques treated words as discrete symbols, ignoring context and similarity, however, it changed with Latent Semantic Analysis (LSA) in which it aimed to uncover the latent conceptual structure between terms and documents through matrix factorization, enabling more meaningful document retrieval (Deerwester et al., 1990). A major advancement came with Word2Vec, which allowed machines to infer relational meaning between terms by predicted a word from its context or vice versa, resulting in embeddings that preserved semantic analogies famously illustrated by $\text{vector}(\text{"King"}) - \text{vector}(\text{"Man"}) + \text{vector}(\text{"Woman"}) \approx \text{vector}(\text{"Queen"})$ (Mikolov et al., 2013). Further refinement was introduced with Global Vectors for Word Representation (GloVe), developed by Pennington et al. (2014) which integrates the strengths of both global matrix factorization techniques like Latent Semantic Analysis (LSA) (Deerwester et al., 1990) and local context window models such as the skip-gram approach proposed by (Mikolov et al., 2013). GloVe employs a global log-bilinear regression model which is an unsupervised method that effectively

captures word co-occurrence information across the entire corpus and has demonstrated superior performance on tasks such as word analogy, semantic similarity, and entity recognition (Pennington et al., 2014).

These embedding techniques are particularly relevant to the research objectives of this study, which seeks to compare LSTM and Transformer architectures in a DBS-specific chatbot. Both architectures fundamentally rely on such vector representations to interpret and generate human-like language, and an important factor in performance outcomes.

After learning the word embedding techniques that encode semantic meaning into dense vector representations (Mikolov et al., 2013; Pennington et al., 2014), the next critical evolution in natural language processing involved architectures capable of modeling sequential dependencies within language (Jurafsky and Martin, 2023). While embeddings improved representation, on their own, they cannot track dependencies across long sentences (like apart from only word, phrases and even sentences using embeddings), an essential requirement for developing coherent and interactive conversational agents (Young et al., 2018).

This limitation led to the emergence of deep learning architectures, particularly Recurrent Neural Networks (RNNs) and their variants such as LSTM and GRU, which became foundational in early neural chatbot systems due to their ability to process data over time (Hochreiter and Schmidhuber, 1997; Cho et al., 2014). Understanding these architectures is crucial for this study, as it sets the stage for a comparative analysis between traditional deep learning models like LSTM and modern Transformer-based models in the development of a DBS-specific chatbot. Furthermore, to address the research question it is essential to first explore the foundational neural network models that support earlier conversational systems.

At their core, neural networks consist of interconnected computational units called neurons that are arranged in feedforward layers without cycles and data is processed by passing it through successive layers to extract hierarchical representations (Jurafsky and Martin, 2009). Feedforward neural networks leverage word embeddings to generalize across semantically similar terms, like even if a model has not seen the phrase “dog gets fed,” it can still make an accurate prediction if it has encountered similar contexts like “cat gets fed,” due to the proximity of embeddings between “dog” and “cat” (Jurafsky and Martin, 2009).

Another innovation allowed cycles in the network architecture which are called Recurrent Neural Networks (RNNs) which enabled information persistence, as the hidden state of the network at each time carries forward learned context from prior steps (Jurafsky and Martin, 2009). RNN-based language models process input sequences word-by-word, updating internal states to predict the next likely word based on both the current input and previously accumulated knowledge (Mikolov et al., 2010). Despite their theoretical ability to retain long-term dependencies, RNNs suffer from vanishing gradient problems, where the influence of earlier inputs diminishes rapidly during training, impairing the model’s ability to retain meaningful long-range context, making it unsuitable for extended sequences, a crucial requirement for maintaining coherent dialogue in chatbot applications (Hochreiter and Schmidhuber, 1997). To overcome these challenges, the Long Short-Term Memory (LSTM) architecture was developed which enhanced RNNs by incorporating specialized memory cells regulated by input, output, and forget gates, allowing them to selectively retain or discard information over long time spans (Hochreiter and Schmidhuber, 1997).

Further innovations such as Gated Recurrent Units (GRUs) simplified the LSTM mechanism while retaining comparable performance, especially in resource-constrained scenarios

(Cho et al., 2014; Chung et al., 2014). These deep learning models enabled systems to produce contextually appropriate and dynamic responses. They have demonstrated the scalability and adaptability of deep learning in dialogue systems (Young et al., 2018; Cambria and White, 2014). Trained on massive corpora, these models significantly outperform earlier rule-based systems in natural language understanding and generation (Al-Amin et al., 2024). However, they also face challenges such as training complexity, sensitivity to noise, and difficulty in retaining long-term coherence issues that have prompted the search for more robust architecture like transformer-based models, (Young et al., 2018; Schuurmans, 2023; Cambria and White, 2014) the focus of this study. Unlike RNNs that process input sequentially, Transformers employ a self-attention mechanism that allows for the simultaneous processing of entire input sequences, innovation that not only improved computational efficiency and scalability but also significantly enhanced the ability to capture long-range dependencies and contextual relationships in text capabilities critical to building coherent and responsive conversational agents (Vaswani et al., 2017).

A key advantage of Transformer models lies in their compatibility with transfer learning where pre-trained language models can be fine-tuned on domain-specific tasks, drastically reducing the need for large, labelled datasets and computational resources (Devlin et al., 2018). This flexibility has made Transformers standard for modern chatbot development, enabling more dynamic, accurate, and context-aware conversational systems (Behl and Bibhu, 2024).

Recent advancements extend beyond traditional Transformer models to address performance and scalability challenges like State-Space Models (SSMs) simplify the Transformer architecture by replacing self-attention and feed-forward layers with linear recurrence mechanisms that efficiently capture both short- and long-term dependencies (Lin & Michailidis, 2024) for faster inference, reduced memory usage, and improved suitability for real-time conversational (Lin &

Michailidis, 2024). Another promising innovation is Memory-Augmented Transformers, which incorporate external memory to retain and retrieve context over extended conversations and enhance coherence in multi-turn dialogues and improve representational capacity Schuurmans (2023). While these approaches are beyond the scope of the current study, they present valuable opportunities for future investigation.

Having reviewed the literature (In the context of this research, which aims to compare Transformer-based and LSTM-based chatbot architectures for a DBS-specific implementation), that highlights not only the technical advances but also the rationale behind transitioning from RNNs to Transformers. The exploration of preprocessing pipelines, embedding methods, and sequential & parallel modeling reveals how each component contributes to the ability of a chatbot to understand and respond meaningfully in real-world use cases. However, given the scale and time bound nature of this work it must be submitted that this is a mere snapshot of all of the potential literature that could be reviewed.

Furthermore, a well-structured study is required to carry out the study to build and compare traditional and newer models in line with the goals of this research. To address this systematically, the next section outlines the research methodology adopted in this study, including the overall approach, design, methods of data collection, and how the data was analyzed to answer the research questions reliably.

3.0 Research Methodology

Research methodology can be referred to as the theory of how research should be undertaken and can be viewed as a multi-stage process that must be undertaken systematically by researchers to produce valid results from reliable data that are aligned with the objective of the

research(Saunders, Lewis and Thornhill, 2009). The stages include formulation of research topic, literature review, understanding the research philosophy, research approach to be undertaken, formulation of research design, addressing ethical issues, data collection(sampling, secondary, observation, semi-structured, questionnaire), data analysis(quantitative or qualitative or mixed), project deliverables like report and presentations(Saunders, Lewis and Thornhill, 2009). These stages represent a perspective about research that presents information in a progressive way from philosophical framework to more specific and detailed procedures (Creswell and Creswell, 2018).

Among these stages, understanding the research philosophy is fundamental, as it shapes the researcher's approach to the entire study from defining the research strategy to data collection and analysis(Žukauskas et al., 2018). The following section discusses various research philosophies and explains the rationale behind the one adopted for this investigation.

3.1 Research Philosophy

In planning a study researcher adopts a research philosophy that contains researcher's thoughts and assumptions about the way he looks at the world (Saunders, Lewis and Thornhill, 2009). In other words, it is the basis of research, which involves the choice of research strategy, formulation of problem, data collection, processing and analysis (Žukauskas et al., 2018).

According to (Creswell, Creswell, 2018) there are 4 main research philosophies, positivism, constructivism, transformative and pragmatism. Positivism as described by (Creswell, Creswell, 2018) is a deterministic philosophy where it is believed that causes likely lead to specific outcome, therefore it often focuses on identifying and evaluating the causal factors that shape the outcome, as seen in the experimental designs. It also aims to reduce the ideas into smaller sets such that it can be tested through hypothesis and research questions. Supporting this view, (Žukauskas

et al., 2018) asserts that the social world can be understood in an objective way, where scientist is an objective analyst and, on the basis of it, dissociate himself from personal values and works independently(Žukauskas et al., 2018). Knowledge generated in this research is built on the measurement of objective reality or human behavior through measurable indicators. Accepted approach to research by positivists is when a formulated theory is either accepted or refuted based on data collection and analysis(Creswell, Creswell, 2018).

Constructivism, on the other hand, is typically associated with qualitative research. Subjective meanings are formed through personal experiences, which are complex and prompt researchers to explore multiple perspectives(Creswell, Creswell 2018). The greatest attention here is given to understanding of the ways through which people experience the social world(Žukauskas et al., 2018). Research questions are often open-ended, allowing researchers to construct broader understanding believed to be influenced by their social or historical background. Instead of starting with a theory (as in positivism) it collects data that either supports or refutes the theory(Creswell, Creswell 2018).

Transformative approach promotes social justice and changes and researchers go beyond understanding the problem and address participants issues like discrimination, inequality and empowerment (Mertens, 2017). Another worldview comes from Pragmatism which doesn't commit to one system of philosophy or reality, and researchers are free to multiple approaches to collect and analyze data rather than subscribing to one(qualitative or quantitative) (Creswell, Creswell 2018). This applies to mixed methods of research & supports ideas from both qualitative or quantitative methods and is more productive. It prioritizes research problems and applies whatever methods are most useful for addressing them. "Numbers and words can work together to

produce richer and more insightful analyses of complex phenomena than could either one alone” (Rossman & Wilson, 1985).

Based on the nature of this study, positivism is the most appropriate philosophy for it. The research involves comparative analysis between the Transformer-based and classical LSTM-based chatbot architectures using quantifiable performance metrics such as match accuracy, inference time, and semantic scores. Since the focus is on objective measurement rather than subjective interpretation of personal beliefs, emotions or social change(due to limited scope of this study), the positivist philosophy aligns well with the study’s aim of performance evaluation in a structured, replicable manner.

Given the adoption of a positive philosophy, the next logical step in the methodological framework is selecting a research approach that aligns with this worldview(**Saunders, Lewis & Thornhill, 2009**). The following section discusses different research approaches in detail and the rationale behind selection of an approach for this study.

3.2 Research Approach

According to (Saunders, Lewis and Thornhill, 2009) the two main research approaches are deduction and induction. These are logical reasoning strategies and guide about how research theory relates to data. With “Deduction” a theory and hypothesis are developed, and research strategy is designed (usually quantitative) to test the hypothesis. Key features include objectivity, control of variables, operationalism (facts to be measured quantitatively), reductionism (breaking down of complex problems.) and generalization(Saunders, Lewis and Thornhill, 2009). While on the other hand, with induction, data is collected (e.g. survey, observations, interviews etc) to build a theory rather than testing an existing one. It is often associated with qualitative studies. Unlike

deduction, it allows flexibility, encourages alternative explanations and develops deeper insights (Creswell & Creswell, 2018).

A positivist stance typically supports a deductive approach, where existing theories are tested through systematic observation and analysis, therefore the approach chosen for this DBS Chatbot study is deductive. In this approach, it begins with a theory or hypothesis and tests it through data analysis (Robson,2002). In this case established theory is that transformers-based architecture has higher efficiency than traditional models like LSTM. The research then tests it by applying both models to the same dataset (DSB specific) to compare the performance metrics. Then the hypothesis will be rejected or accepted based on the tests making it a deductive approach. This approach is particularly appropriate in this study as there is already established literature providing evidence that Transformers outperforms traditional models(Vaswani et al., 2017) and the same can be tested in the context of DBS chatbot. Deductive research approach applied to DBS specific chatbot study can be visualized as

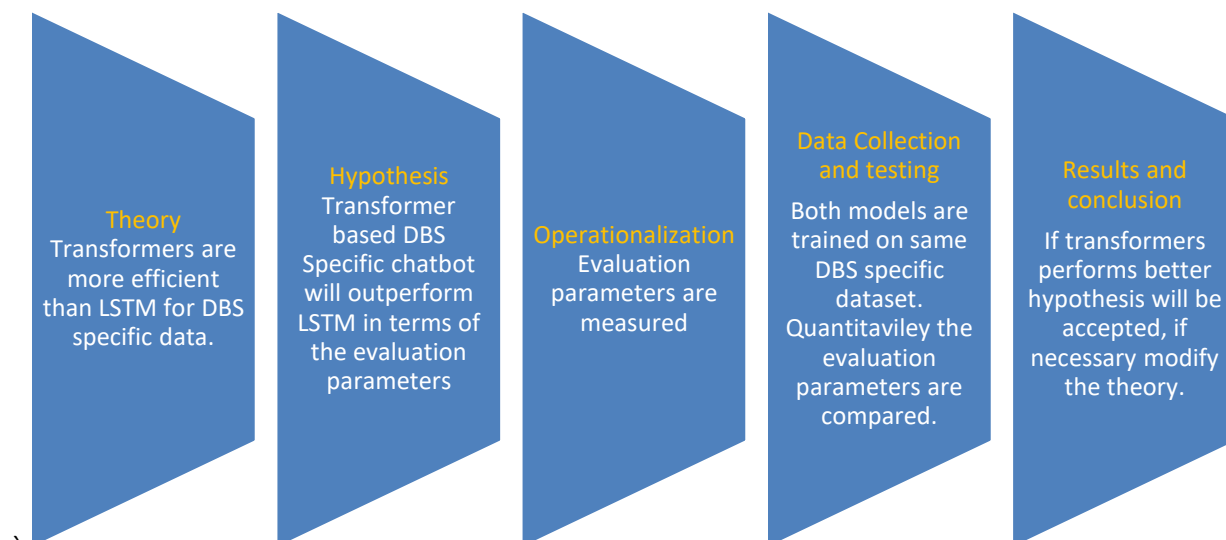


Figure 1 Five Stages of Deductive Research(Robson,2002)

Having determined the research philosophy and approach, the next logical step is to formulate a structured plan to execute the study. This is referred to as the research design and it ensures that the methods and procedures used align with the research objectives and philosophical stance (Saunders, Lewis and Thornhill, 2009). The following sections lay emphasis on the different design aspects and rationale behind choosing the appropriate design for this study.

3.3 Research Design

According to (Robson, 2002), Research design can be considered as a process or general plan that turns research question into research project. It consists of three layers, research strategies, research choices and time horizons (Saunders, Lewis and Thornhill, 2009). The different research strategies are experiments, survey, case study, action research, grounded theory, ethnography & archival research. Time-horizons answers the question of whether research is a snapshot taken at a particular time (cross-sectional) or a series of snapshots and representation of events over a period of time (longitudinal) (Saunders, Lewis and Thornhill, 2009).

In this research the best fitting design is Experimental study as it involves study of change in performance of models in terms of accuracy, speed, etc. Furthermore, the research follows a cross-sectional design, meaning that the data is collected at a single point in time, for evaluation of dataset. The design is appropriate as study evaluates the performance of both the Transformer and classical models at a particular stage and no longitudinal analysis over time is required.

Having established the research design, it is important to define the specific methodological approach that guides the data collection and analysis & interpretation process involving several decisions that need to be taken in order (Creswell & Creswell, 2018). This helps determine whether a qualitative, quantitative, or mixed methods approach is most suitable (Creswell & Creswell,

2018; Saunders, Lewis and Thornhill, 2009). In line with these considerations, the next section outlines the different research methods and rational behind adopted method for this research study.

3.4 Research Method

Research approaches in a methodological sense (methodological approaches) are categorized into Qualitative, quantitative and mixed methods (Creswell & Creswell, 2018; Saunders, Lewis and Thornhill, 2009). Quantitative approaches can be viewed as examination of variables which can be measured typically on instruments and numerical data that can be analyzed statistically while qualitative research is an approach where, rather than numbers, it uses non numerical forms of data like words, observations, and explores understanding or meaning from how individuals or groups interpret a social or human issue (Creswell & Creswell, 2018; Saunders, Lewis and Thornhill, 2009). Mixed methods leverage the strengths of both qualitative and quantitative methods by integrating numerical data and comprehensive understanding of the data(Creswell, Creswell 2018).

This study adopts a mono-method (single data collection technique) quantitative approach focusing exclusively on numerical comparisons and statistical analysis. Because of its limited scope, it does not incorporate qualitative methods such as interviews, surveys or subjective analysis. It involves training and testing different models (Transformer and LSTM) on the same DBS-specific dataset and focuses purely on comparing the performance metrics of the two comparisons. As this is an applied research project, the primary aim is to directly compare how the models perform in a practical, real-world environment rather than exploring theoretical concepts. Therefore, the research method is quantitative, as it seeks to derive insights from data.

Following the selection of research methods and approach, it's helpful to take a step back and consider the broader purpose of this study. Research can generally be classified based on whether it aims to solve real-world problems(applied) or contribute to academic(basic) which helps clarify what the study is trying to achieve and who it is intended to benefit (Robson,2002). The following section explores these types of research and explains why this particular study falls under the applied category, given its practical focus on improving chatbot performance for student support.

There are various types of research in existence. Some of them mentioned by (Rossman & Wilson, 1985) Applied(Real world) research and Basic(Academic) research. The applied research focuses on solving practical problems and aims to understand lived experiences and generate actionable insights relevant to daily life and practice (Rossman & Wilson, 1985). It is relatively small-scale research carried out by individuals or small teams. In contrast, basic research is mostly concerned with developing and extending theoretical knowledge. Such research is largely undertaken in universities and largely as the result of an academic agenda(Saunders, Lewis and Thornhill, 2009).

Since this study is centered on practical and real-world evaluation of chatbot architectures to improve student support services it clearly falls under the category of applied research. It adopts a positive research philosophy in a deductive manner, testing an established theory by Vaswani et al., (2017) that Transformer models outperform LSTM in efficiency and accuracy. The research follows a cross-sectional design and uses a mono-method quantitative approach to compare model metrics such as match accuracy, inference time, and semantic scores. The outcome will be a functional chatbot artefact developed using Python, Google Colab, and HuggingFace, that attempts to offer scalable and effective solutions within educational institutions.

Having established the research methodology and before we begin to collect the data, it is equally essential to give serious thought to the ethics of what you are proposing (Robson, 2002). Ethics are the principles and guidelines that help us to distinguish between right and wrong and to do the right thing and assist researchers in conducting ethically sound research studies (Johnson & Christensen, 2000). In the context of this study, which involves the handling of digital data and potential user interactions through a chatbot interface, ethical considerations are imperative. The following section outlines the ethical considerations considered to ensure the study is conducted in a moral and responsible way and adheres to best practices in research ethics.

4.0 Ethical considerations

According to Saunders, Lewis and Thornhill (2009), ethical issues can arise at any stage of a research project particularly while collecting data from online sources such as websites, e-books, and datasets. In the context of this study where it involves use of datasets potentially derived from open educational platforms, such as DBS web pages and institutional brochures, In accordance with the EU General Data Protection Regulation (GDPR), care was taken to ensure that all data used complies with legal and ethical standards (A link to the GDPR guidelines is provided in reference #38).

Since the study was conducted within DBS context, attention was given such that no DBS faculty related data was retained, and sensitive details such as faculty names, contact numbers, and emails were removed during data preprocessing and cleaning step. Additionally, cleaned data is used during the model training and deployment on platforms such as Google Colab and Hugging Face, to maintain data security.

As noted by Johnson and Christensen (2000), Conducting research through the Internet medium also raises ethical issues around topics such as informed consent & privacy of data. However, since the data in this study is publicly accessible and does not involve direct human participants, informed consent was not required.

This research also aligns with the European Union's Artificial Intelligence Act (2024), which emphasizes the importance of maintaining technical documentation that outlines the AI system's purpose, design, and interface. A link to the Act is provided in reference #39. To ensure this, appropriate documentation has been maintained throughout the chatbot development to meet these best practices.

Since this study involves Natural Language Processing(NLP), there are some issues that lead to ethical problems in NLP, described by Bender et al. (2020) as dual use and bias. Bias can result from biased datasets or under or oversampling of the data, resulting in inaccurate chatbot response, while dual use refers to when system (here chatbot) is used in harmful ways (Bender et al, 2020). To mitigate these risks, datasets are curated from publicly accessible data that is DBS specific as well as generic university student related queries and responses and like universal greetings or goodbye messages. The prediction of the response is not based on particular groups like genders, age or region, however offensive language will be detected and filtered, therefore, no further step is required to ensure that the product is used so that it causes any harm to anyone.

Furthermore, in presenting research outcomes like project reports, transparency and replicability should be observed (Johnson & Christensen, 2000). In this research, while writing the report, no information is fabricated or falsified. Proper reference is used when making the use of

the contributions of others thus committing to transparency & proper research methods are recorded to allow reproducibility by other researchers.

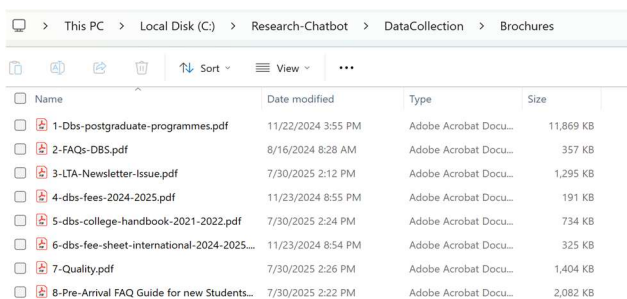
By adhering to ethical guidelines, this research aims to build accountability and trust in its results.

After ensuring ethical standards are met, the next step in research involves collection and processing of data(Saunders, Lewis and Thornhill, 2009).

5.0 Data Collection and Analysis

According to Saunders, Lewis and Thornhill (2009), there are two main types of data Primary and Secondary. Primary data is collected directly by researchers for their studies addressing specific research objectives, on the other hand, Secondary data is already collected data for some other purpose and can be used in research purpose like documentary data, survey data etc. This research uses secondary data obtained from publicly accessible sources.

DBS brochures – Collected Dublin Business School publicly available brochures(like websites, student portals, emails etc) in pdf formats.



Name	Date modified	Type	Size
1-DBS-postgraduate-programmes.pdf	11/22/2024 3:55 PM	Adobe Acrobat Docu...	11,869 KB
2-FAQs-DBS.pdf	8/16/2024 8:28 AM	Adobe Acrobat Docu...	357 KB
3-LTA-Newsletter-Issue.pdf	7/30/2025 2:12 PM	Adobe Acrobat Docu...	1,295 KB
4-dbs-fees-2024-2025.pdf	11/23/2024 8:55 PM	Adobe Acrobat Docu...	191 KB
5-dbs-college-handbook-2021-2022.pdf	7/30/2025 2:24 PM	Adobe Acrobat Docu...	734 KB
6-dbs-fee-sheet-international-2024-2025...	11/23/2024 8:54 PM	Adobe Acrobat Docu...	325 KB
7-Quality.pdf	7/30/2025 2:26 PM	Adobe Acrobat Docu...	1,404 KB
8-Pre-Arrival FAQ Guide for new Students...	7/30/2025 2:22 PM	Adobe Acrobat Docu...	2,082 KB

Figure 2 - The collection of DBS brochures

Web Content Scraping – Performed Web scraping of the DBS website pages using python library BeautifulSoup to extract the information in text format. Consolidated all the web pages content

into a single web scraped corpus. Please refer to appendix A for a sample conversion. Once the data is collected next step in the pipeline is to clean and perform some preprocessing before it can be used by models for training and testing.

```
Why DBS
urls = [ "https://www.dbs.ie/dbs-staff", "https://www.dbs.ie/about-dbs/welcome", "https://www.dbs.ie/about-dbs/strategic-plan",
"https://www.dbs.ie/about-dbs/history-of-dbs", "https://www.dbs.ie/about-dbs/recognition-and-accreditation", "https://www.dbs.ie/about-
dbs/dbs-institutional-profile-2024", "https://www.dbs.ie/about-dbs/transnational-programmes-at-dbs", "https://www.dbs.ie/about-dbs/tuition-
delivery", "https://www.dbs.ie/about-dbs/for-employers", "https://www.dbs.ie/about-dbs/for-guidance-counsellors",
"https://www.dbs.ie/about-dbs/kaplan-dublin-business-school", "https://www.dbs.ie/about-dbs/jobs-dublin-business-school",
"https://www.dbs.ie/about-dbs/hetac-institutional-review", "https://esource.dbs.ie/home", "https://www.dbs.ie/about-dbs/why-dbs",
"https://www.dbs.ie/about-dbs/athena-swan", "https://www.dbs.ie/about-dbs/gender-pay-gap" ]

v Student experience

urls = [ "https://www.dbs.ie/dbs-staff", "https://www.dbs.ie/about-dbs/welcome", "https://www.dbs.ie/about-dbs/strategic-plan",
"https://www.dbs.ie/about-dbs/history-of-dbs", "https://www.dbs.ie/about-dbs/recognition-and-accreditation", "https://www.dbs.ie/about-
dbs/dbs-institutional-profile-2024", "https://www.dbs.ie/about-dbs/transnational-programmes-at-dbs", "https://www.dbs.ie/about-dbs/tuition-
delivery", "https://www.dbs.ie/about-dbs/for-employers", "https://www.dbs.ie/about-dbs/for-guidance-counsellors",
"https://www.dbs.ie/about-dbs/kaplan-dublin-business-school", "https://www.dbs.ie/about-dbs/jobs-dublin-business-school",
"https://www.dbs.ie/about-dbs/hetac-institutional-review", "https://esource.dbs.ie/home", "https://www.dbs.ie/about-dbs/why-dbs",
"https://www.dbs.ie/about-dbs/athena-swan", "https://www.dbs.ie/about-dbs/gender-pay-gap" ]

contact us

urls = [ "https://www.dbs.ie/about-dbs/contact-us", "https://www.dbs.ie/about-dbs/location", "https://www.dbs.ie/enquirv/enquire-online",
"https://www.dbs.ie/international-students", "https://www.dbs.ie/international-students/european-students/one-semester-options",
"https://www.dbs.ie/international-students/european-students/two-semester-options", "https://www.dbs.ie/international-students/european-
students/how-to-apply", "https://www.dbs.ie/international-students/european-students/accommodation-for-eu-students",
"https://www.dbs.ie/international-students/european-students/tuition-fees", "https://www.dbs.ie/contact-dbs/visit-dbs" ]

news and events

urls = [ "https://www.dbs.ie/about-dbs/news-and-events", "https://www.dbs.ie/about-dbs/news-and-events/2025/06/06/parents-guide-to-
cao-application-form", "https://www.dbs.ie/about-dbs/news-and-events/2025/06/04/cao-change-of-mind", "https://www.dbs.ie/about-
dbs/news-and-events/2025/01/28/dbs-becomes-the-first-independent-hetac-to-complete-the-cinnte-review", "https://www.dbs.ie/about-
dbs/news-and-events/2025/01/10/dbs-athena-swan-bronze-award", "https://www.dbs.ie/about-dbs/news-and-events/2024/12/04/dbs-
becomes-the-first-educational-provider-in-ireland-to-join-the-hidden-disabilities-sunflower" ]
```

Figure 3 - Collection of DBS Web URLs for web scraping

5.1 Data Cleaning and Preprocessing

1. **PDF to text conversion** – The collected brochures were originally in pdf formats contained complex layouts, including tables, images, and multi-column structures, which made direct text extraction difficult. To simplify this process, **Docling** a python library is used which converts documents(see appendix A for example demonstration) of different formats into a unified internal structure which allows it to parse and serialize content into clean, plain text. Thus, content from all the brochures was extracted consistently through same technique.

2. **Text Cleaning and Normalization** – Special characters, extra whitespaces, html tags, numbering and other irrelevant characters were removed from both the brochure and Web scraped data files. Content is lowercased and tokenized for further processing.
3. **Intent Datafile creation** – Manually created an intent file in json format for generic conversational intents like greeting, farewell, thanks etc. along with sample patterns and responses. This will be used in chatbot to check if the user query is about a generic intent. If yes, then get the response from one of the responses corresponding to the predicted intent. Else use the pre-trained mode.
4. **Datafile for testing** – Generated a separate test json file in question answer format consisting of user queries and correct responses based on the brochure\web scraped data. This file will be further used in the evaluation pipeline of the chatbot.
5. **Tokenization** – In order to encode each word into an id, tokenization technique is used.
 1. IN LSTM NLTK library's word tokenization is used
 2. In Transformer based version, model's native tokenizer is used where tokenization happens implicitly.
6. **Vectorization / Embeddings** – In order to convert the natural language text into numerical form to be processed further by models, both the transformer and LSTM versions employ different embedding strategies.
 - a. **Transformers** – In transformers version the embedding is achieved through “all-MiniLM-L6-v2” SentenceTransformer model from the Hugging Face library. It maps sentences & paragraphs to a 384-dimensional dense vector space capable of capturing semantic similarities. Paragraph in each corpus is embedded into a fixed length vector and indexed using FAISS, a similarity search library (it contains

algorithms that search in sets of vectors of any size). When inferred, incoming user query is encoded into same encoding space and compared against the stored vectors.

- b. **LSTM** – In this a static vectorization technique is used. A vocabulary is first constructed from lowercased and normalized training data though NLTK library. Each word is assigned a unique id and encoded into vector by an embedding layer initialized randomly and trained though training data. Questions are mapped into fixed length tokens(20) and passed through single layer LSTM encoder with 128 hidden units. The final hidden state (fixed dimension) was used as the dense representation of the query. For retrieval, cosine similarity was measured between the embedding of the user query and the precomputed embedding of all training questions.
7. **Label & Dictionary Creation** - This stage focuses primarily on the creation of a word-to-index dictionary necessary to translate textual inputs into the numerical domain required for LSTM embedding and sequence processing. All unique words present in the lowercased training questions are extracted and then their frequency is counted. Each word is then mapped to a unique integer index. This dictionary allows the encoding function to convert each input question into a sequence of integer Id. In transformers this process is not explicitly required, as it depends on the vocabularies of two pretrained Transformer models: the SentenceTransformer encoder (all-MiniLM-L6-v2) and the T5 generative decoder (t5-base)

Once the data is checked for errors and cleaned and initial preprocessing is done, data analysis phase can begin (**Saunders, Lewis and Thornhill, 2009**). According to these authors, Exploratory

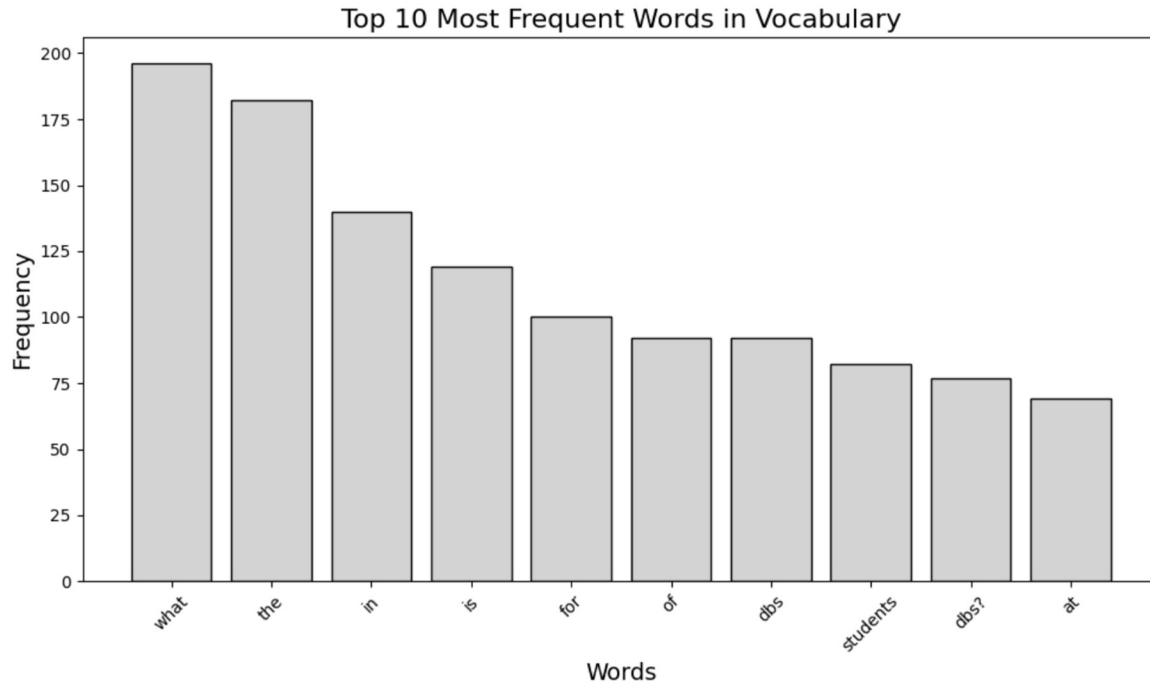


Figure 5 - Bar chart showing the frequency of words.

Overall, EDA analysis verified that the dataset possesses sufficient word diversity to train the models and ensures that the distribution of topics across the corpus aligns with the intended application scope of the chatbot(education and DBS specific).

5.3 Model Development

The project has two separate modelling for transformers and LSTM. The transformer adopts pretrained Retrieval Augmented Generation setup, in which a SentenceTransformer “all-MiniLM-L6-v2” is used to produce 384 dimensional embeddings for both user queries and corpus paragraphs. These vectors are indexed into a FAISS (search library) similarity index for efficient nearest-neighbor retrieval. The generative component is initialized using “t5-base” encoder-decoder google model. The user query and retrieved passages are tokenized using t5 and passed into the t5 model to generate the relevant generative answer.

LSTM follows an approach in which vocabulary is created which initializes an embedding layer with vector size 100. A single-layer Long Short-Term Memory (LSTM) network containing 128 hidden units is defined via the custom LSTMEncoderClass. Each input question (both from the corpus and from users) is passed through the LSTM encode. Embeddings are computed once for all known training questions and stored in `question_vecs_tensor`. The cosine-similarity is used for retrieval during real-time interaction.

In transformer version, the user's query is converted into a dense vector embedding. The pretrained SentenceTransformer encoder (all-MiniLM-L6-v2) is used to do this transformation. This embedding is then submitted to an FAISS index after conversion. In order to make an estimated nearest-neighbor search easier, the index is created during the preprocessing stage. Following this process, the index generates the original text passages for the top three semantically related passage vectors ($k=3$). This approach enables the machine to produce conversationally fluid language.

User Query \rightarrow SentenceTransformer.encode \rightarrow FAISS retrieval \rightarrow T5 generation \rightarrow Output

In LSTM version user query is first converted into lowercased text, then tokenized (converted into integers ids). Then the sequence is padded (to max 20) and passed into LSTM encoder which is treated as final embedding vector. Cosine similarity scores are calculated between this query and the previously computed embeddings of all training questions. Index with the highest similarity score is selected. The rule is if the score exceeds the threshold value (let's say 0.6), the corresponding answer from the training corpus is returned. Otherwise, system provides a standard reply that Sorry, I don't understand.

User Query \rightarrow tokenize \rightarrow encode_question \rightarrow LSTM \rightarrow cosine similarity \rightarrow Retrieval \rightarrow Output

A key implementation detail is that both LSTM and Transformer models were saved locally as well as pushed to Hugging Face library hub. This is to ensure that the models can be reused without retraining, this also enables the chatbot UI to load pretrained models dynamically. Reusing the pretrained and saved model saves lot of time and query response is generated without any delay to the user. Please refer to Appendix B for visualization of different phases in the models' development.

5.3 System Design and Implementation

In order to operationalize the DBS Chatbot, a system was designed that integrates a user interface with a text and voice-based interaction buttons, a backend pipeline to process user entered or spoken query and generate response. The overall flow of the system is depicted below.

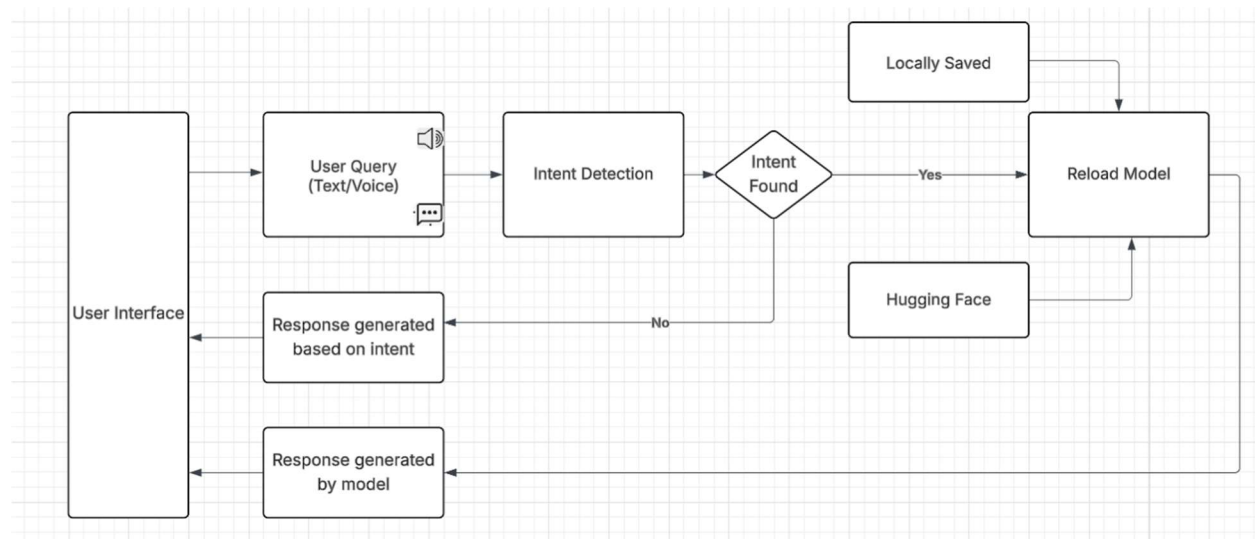


Figure 6 - Query processing flow.

System follows a layered architecture that separates presentation layer with the application logic layer. Following modules constitutes the overall architecture of the system

1. DBSChatbotInterface.py – This module implements chatbot User interface using TKinter library. The onclick and on-pressed events are captured that passes the user text\voice into the backend layer. The events are also responsible for displaying the returned response from the backend. The user interface looks like below.
2. PredictResponse.py – It possesses the backed logic. Based on the received query from the user interface, it first detects the intent using intent classifier, specifically for greetings, thanks, and farewell messages. If intent is matched a predefined response is returned to the user based on pattern matching. Otherwise, it loads pretrained transformer model.
3. VoiceBasedLogic.py - This module handles voice-based interaction.

Please refer to appendix A for a sample text based and voice-based interaction from the user interface and appendix B for listing of all the libraries, files, repositories, readme instructions linked to each module with snapshots.



5.4 Model Evaluation and Benchmark Comparison

Both the chatbot architectures are evaluated using same test file constructed in question-and-answer form in json format and to quantify the performance, four quantitative measurable metrics are used. Exact Match Accuracy measures the proportion of responses that exactly matches the standard answer. It is the strictest parameter because it gives the highest level of correctness of the test. Another important metric is BLEU (Bilingual Evaluation Understudy) which evaluates n-gram similarity between generated and reference answers. In other words, it judges the quality of the translation of the generated answer as compared to the expected one (Papineni et al., 2002). BLEU is simple and general measure and rely on simply n-gram overlap counts between the candidate and referenced sentences, however it fails to account for semantic equivalence (Zhang et al., 2020). Therefore, another metric BERTScore-F1 proposed by Zhang et al., 2020 is

employed to compute similarity in a contextual embedding space using a pretrained BERT model, thereby offering a more robust measure of semantic alignment. Above metrics focusses on the quality of the language only, therefore in order to measure computational efficiency another metric Average Response Time is used to record the inference latency per query. Collectively, these metrics attempt to enable a comprehensive coverage of different aspects of the evaluation of a chatbot system.

LSTM-based Retrieval Model achieved exact match accuracy of 0.45 which indicates that 45% of the responses generated match exactly the expected answers in the test file. In other words, more than half of the answers could not be matched exactly with the expected answers. The Average BLEU score of 0.21 indicates a limited level of n-gram overlap between generated and referenced answers. In other words, the translation of the expected answers was not good in quality and had difficulty handling the paraphrasing. Surprisingly, the BERT F1 score of 0.87 indicates that the semantic alignment with the generated answers is quite high. The average inference time was close to 0 signifies its strength in speed and fast lookups. It implies that this architecture is lightweight and might be effective for smaller datasets and latency sensitive deployments but will not be effective in conversational systems.

```

➡ Some weights of RobertaModel were not initialized from the model checkpoint at rob
You should probably TRAIN this model on a down-stream task to be able to use it fo
calculating scores...
computing bert embedding.
100%  5/5 [01:23<00:00, 12.01s/it]
computing greedy matching.
100%  3/3 [00:00<00:00, 25.29it/s]
done in 83.30 seconds, 2.06 sentences/sec

--- Evaluation Summary ---
Accuracy (Exact Match): 0.45
Average BLEU Score: 0.21
Average BERTScore F1: 0.87
Average Response Time: 0.00s

```

Figure 7 - LSTM model evaluation result.

In contrast the Transformer Retrieval-Augmented Generation model demonstrated stronger performance on most of the metrics used in evaluation. It achieved exact match accuracy of 84%, indicating its ability to not only to retrieve the contextually correct text but also generate a well-formed answer. Both the BLEU score(0.68) and BERT F1 score (0.96) indicate superior semantic similarity to referenced answers and support the advantages of context-aware embeddings over sequence-to-sequence generation. However, the latency is the tradeoff for the increase in the quality of answer generating in transformers. Inference time in this case is 1.44 seconds per query which is quite high as compared to the LSTM.

```

Evaluating Query 1/2/1/2
Some weights of RobertaModel were not initialized from the model checkpoint at roberta-large and are newly initialized: ['pooler.de
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
calculating scores...
computing bert embedding.
100% ██████████ 3/3 [00:41<00:00, 12.02s/it]
computing greedy matching.
100% ██████████ 3/3 [00:00<00:00, 37.94it/s]
done in 41.29 seconds, 4.17 sentences/sec

--- Evaluation Summary ---
Accuracy (Exact Match): 0.84
Average BLEU Score: 0.68
Average BERTScore F1: 0.96
Average Inference Time: 1.44 seconds

[{'Query': 'What is PSI?',
  'Generated': 'to promote and support psychology education and practice in Ireland Higher Diploma in Arts in Psychology',
  'Expected': 'professional body for psychology in Ireland',
  'ExactMatch': 0,
  'BLEU': 0.04030833724870907,
  'TimeTaken': 1.9852635860443115},
 {'Query': 'What country has the highest reputation for leading independent colleges?',
  'Generated': 'Ireland',
  'Expected': 'Ireland',
  'ExactMatch': 1,
  'BLEU': 1.0,
  'TimeTaken': 0.5663714408874512},
 {'Query': 'student population of DBS?',
  'Generated': 'over 9,000',
  'Expected': 'over 9,000',
  'ExactMatch': 1,
  'BLEU': 1.0,
  'TimeTaken': 0.5663714408874512}]

```

Figure 8 - Transformer model evaluation result.

Below is the graphical representation of above comparison.

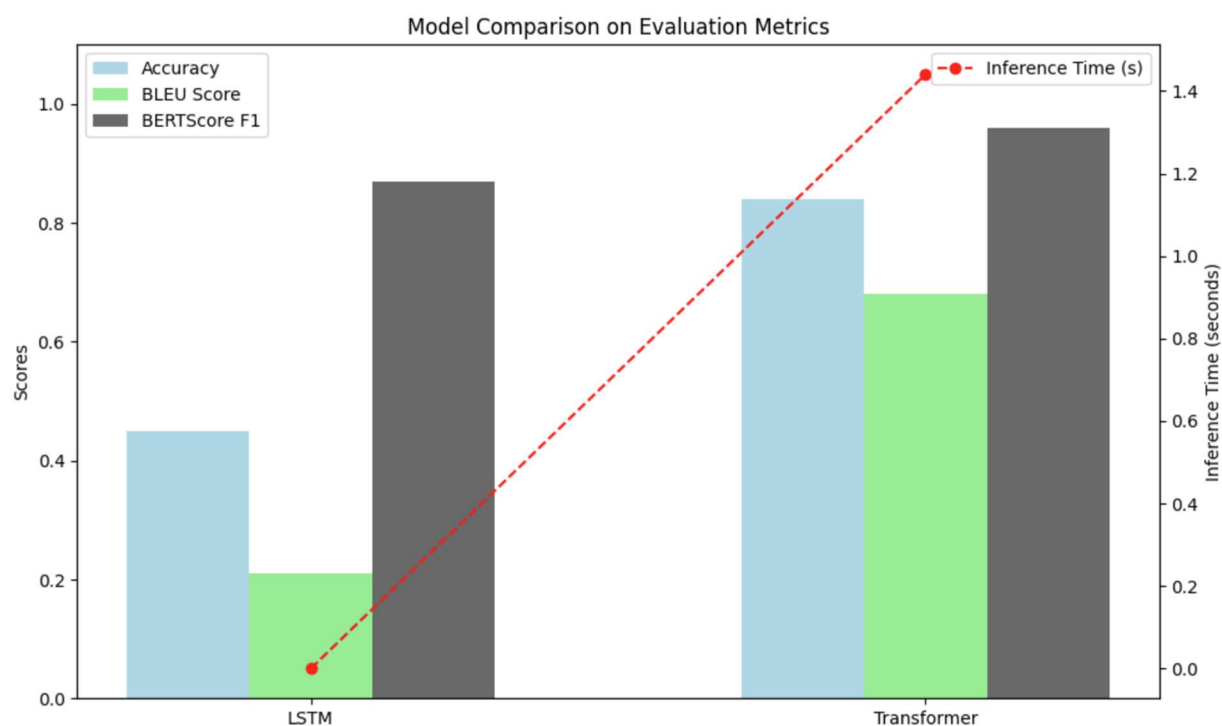


Figure 9 - Comparison plot of the model evaluation results.

This superior performance of the transformer can be attributed to the architectural advantages of the Transformer. In contrast to LSTMs which are sequential and suffer from the vanishing gradient problem in long-range dependencies, the Transformer applies attention mechanisms to capture global relationships among words in a sentence (Hochreiter and Schmidhuber, 1997). This enables the model to weigh the importance of each token with respect to the others which ensures better semantic understanding, response prediction (Behl and Bibhu, 2024). In addition, Transformers can process sequences in parallel, which can lead to more efficient computations, and can also take advantage of pre-trained embeddings extracted from large-scale corpora, that heavily boosts generalization to unobserved queries. All of these factors account for why the Transformer-based chatbot outperformed the LSTM model in producing semantic responses according to the aforementioned theoretical and empirical studies.

All the evaluation parameters data for each query can be plotted to check their distribution. This will further assist in the hypothesis testing where testing techniques are selected based on the normality of the distributions.

Exact match distribution shows plot below shows the distribution of the exact match accuracy of each query in both the LSTM and Transformer cases.

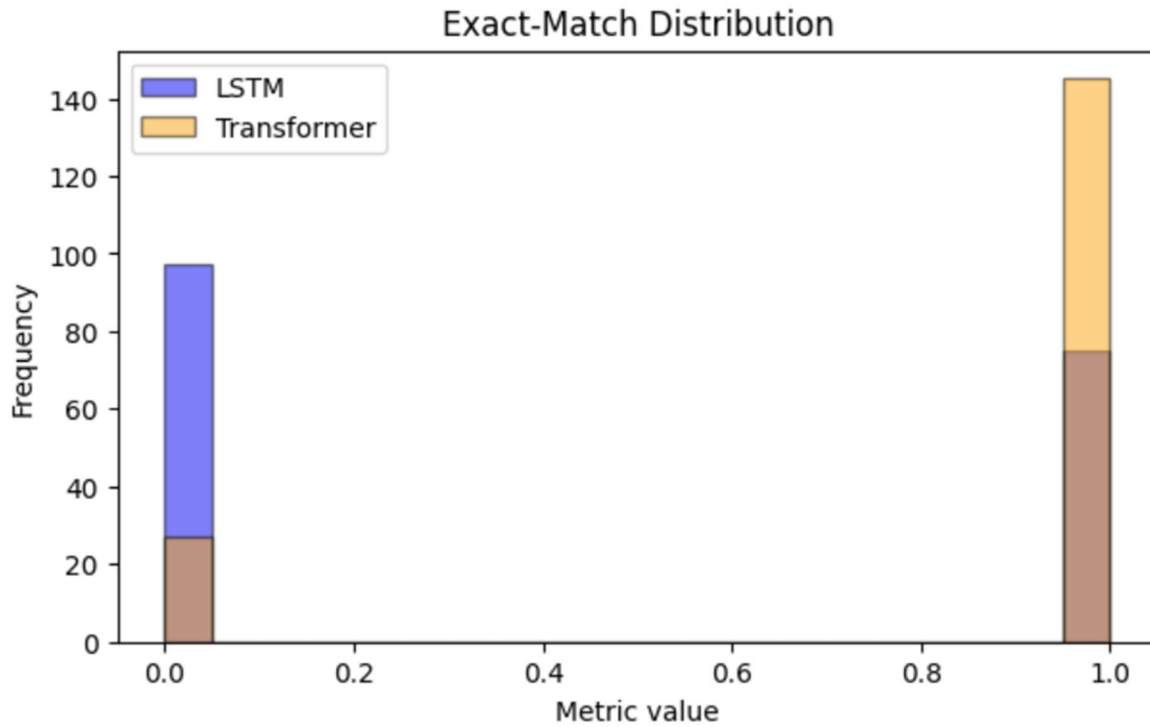


Figure 10 - Exact match metrics distribution chart.

BLEU distribution shows the distribution of the BLEU score of each query in both the LSTM and Transformer cases. It also indicates that the data is not normally distributed(not a bell curve shaped).

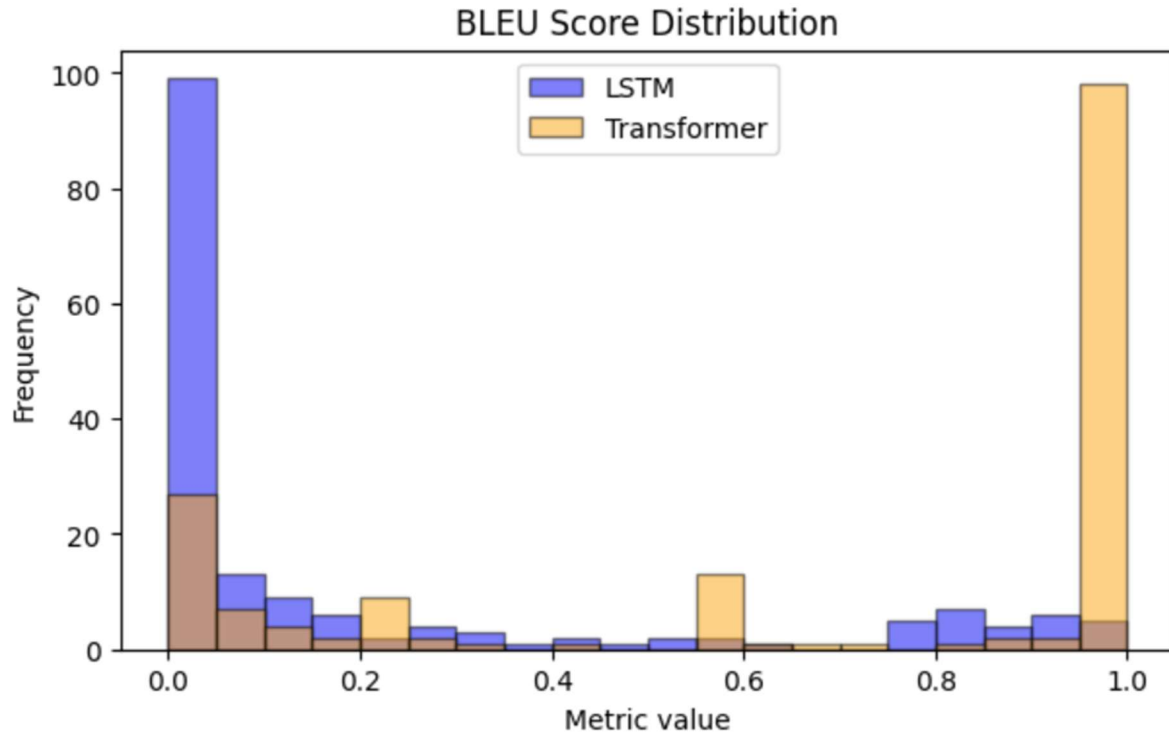


Figure 11 - BLEU Score metric distribution chart.

BLEU distribution shows the distribution of the Inference time score of each query in both the LSTM and Transformer cases. It also indicates that the data is not normally distributed(not bell curve shaped).

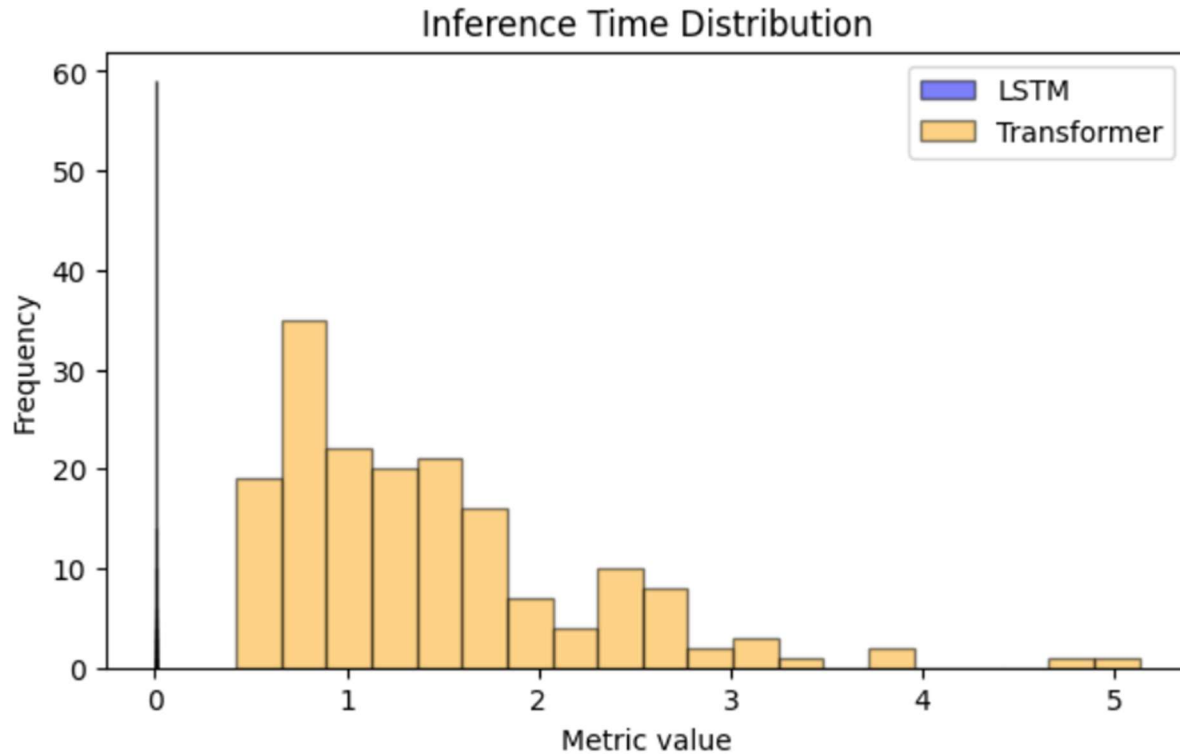


Figure 12 - Inference time metric distribution chart.

The visual representations of the above plots suggest that the data is not normally distributed. Furthermore, in order to statistically compare the two sets of data, one that is being collected and other that is theoretically expected, process known as hypothesis testing is used (Saunders, Lewis and Thornhill, 2009).

Hypothesis testing

To address the research question — *“How can transformer-based architecture improve efficiency and accuracy of a DBS-specific chatbot compared to traditional models like LSTM?”*

A hypothesis test was conducted to attempt to compare LSTM and Transformer architectures trained on the DBS-specific dataset. The established theory, supported by prior literature

(Vaswani et al., 2017), suggests that Transformer-based architectures outperform traditional models like LSTM in both efficiency and accuracy. Hypothesis drawn is

Null Hypothesis (H_0): *The Transformer-based architecture does not perform better than the LSTM-based architecture on the DBS chatbot task (i.e., there is no statistically significant difference in Accuracy, BLEU Score, or BERTScore between the two models).*

To test this hypothesis, both models were applied to the same dataset, and key performance metrics Accuracy (Exact Match), BLEU score (Papineni et al., 2002), and inference time were computed per question. After that in order to check the normality of the data, A Shapiro–Wilk test for normality was applied, which indicated non-normal distributions ($p < 0.05$). Since the data was not normally distributed, Wilcoxon signed-rank test (Wilcoxon, 1945), was conducted for all the metrics, which returned a very small p value of 0.001. According to this test, a p-value of less than 0.001 supports the rejection of null hypothesis that the two models perform equivalently. This further supports that the transformer-based architecture performed efficiently and accurately as compared to LSTM in this research. It is also important to emphasize that this hypothesis testing is performed under the limitations of a DBS-specific dataset, without qualitative aspects (due to time bound nature of the study), a focused literature base, and a restricted testing scale therefore, while the results support the theory that Transformer architectures outperform LSTMs, they do so only within the confines of this controlled experimental setup.

```

.....Hypothesis Testing .....
Null Hypothesis: The Transformer-based architecture does not perform better than the LSTM-based architecture on
the DBS chatbot task (i.e., there is no statistically significant difference in Accuracy, BLEU Score, or BERTScore between the two models)
Alternative Hypothesis: The Transformer-based architecture does perform better than the LSTM-based architecture on the DBS chatbot task

First Checking for normality of the data using Shapiro test and then based on it, hypothesis test will be chosen

Exact Match = Normality p-value (Shapiro): 0.0000
Differences NOT normally distributed: using Wilcoxon signed-rank test
Wilcoxon p-value = 0.0000
Reject null hypothesis

BLEU = Normality p-value (Shapiro): 0.0000
Differences NOT normally distributed: using Wilcoxon signed-rank test
Wilcoxon p-value = 0.0000
Reject null hypothesis

Inference Time = Normality p-value (Shapiro): 0.0000
Differences NOT normally distributed: using Wilcoxon signed-rank test
Wilcoxon p-value = 0.0000
Reject null hypothesis

```

Figure 13 - Hypothesis testing output.

6.0 Conclusion

This study attempted to demonstrate the potential advantages of transformer-based architecture over traditional LSTM models in developing a DBS specific chatbot that supports both text and voice-based interactions. To achieve this, two objectives were outlined, and they were to allow implementation and demonstration of comparable chatbot versions leveraging Transformers and LSTM models. Both the architectures model was trained & evaluated with the same DBS specific data and quantitative measurable metrics followed by a hypothesis testing. The findings showed that the Transformer architecture delivered better accuracy, semantic similarity and response quality as compared to LSTM, however with the trade-off of higher inference time. These findings agree with the established theories of the Vaswani et al., 2017 and Behl and Bibhu, 2024, however this research had a restricted scope. The study primarily focused on quantitative evaluation through measurable metrics such as Exact match accuracy, BLEU and BERT scores and inference times. Due to limited scope and time period of this research, future studies can be extended by incorporating qualitative aspects like emotions and user experience to further boost the performance and relevance of the chatbot conversation. The study can be extended to ensure it

is highly scalable and relevance to schools and institutions beyond DBS, aiming to enrich student engagement, support services, and automate query resolution using cutting-edge AI technologies.

7.0 Limitations and Future Scope

While the study supports the fact that potentially the transformers-based architecture performed better than LSTM and suits their implementation in the conversational assistants, several limitations must be acknowledged. This study was conducted on relatively small DBS specific dataset. Its scope is limited and may limit the generalizability of the findings. Evaluation of the results are also confined to quantitative metrics such as exact match accuracy, semantic metrics such as BLEU and BERT scores and inference time, while qualitative aspects like user experience, satisfactory scores, feedback, emotional engagement etc were not assessed due to time bound nature of the project. Furthermore, the literature review and testing scaled were limited. Additionally, hardware constraints restricted experimentation, which may have further improved performance but required greater computational resources.

Future research can address these limitations by expanding datasets to include data from different universities , in different languages for diverse contexts, different data collection methods including surveys for qualitative analysis, more advanced transformer architect can be used for increased performance, effective deployment techniques for scalability, pipelines to train the models continuously over a period of time, for robust architecture.

8.0 Appendix A Sample Implementation and Examples

Docling Example showing how Docling library can be utilized to convert complex formats into simpler text so that it can be further fed to the models. Here, a sample Dbs-postgraduate-programmes-pdf, a 96 page 11.6 MB file obtained from the DBS website is used. Its layout shows the pictures, tables, frames etc from which text needs to be extracted. Docling library converts the different formats of documents into Docling's unified fundamental document representation and then serializes the content to the desired simplified text format.

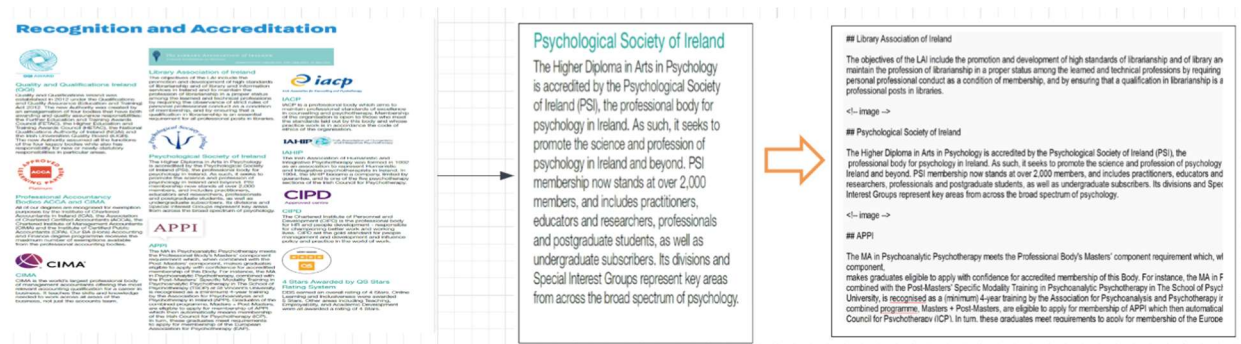


Figure 14 - pdf into text conversion through docling

Web Scraping - is a technique which scrapes the data from web pages and can be used to build a corpus. There are various libraries available including BeautifulSoup, which is utilized in this study to scrape the DBS web site pages into simple text data. Below is the screen shot of the data obtained from the <https://www.dbs.ie/about-dbs/contact-us> link.

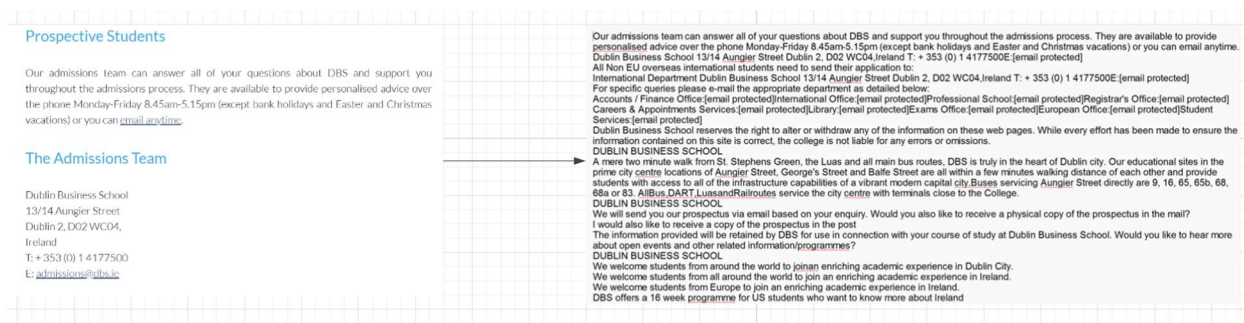


Figure 15 web scraping of DBS contact-us web page

Sample Chatbot Interaction - in text mode through send button.

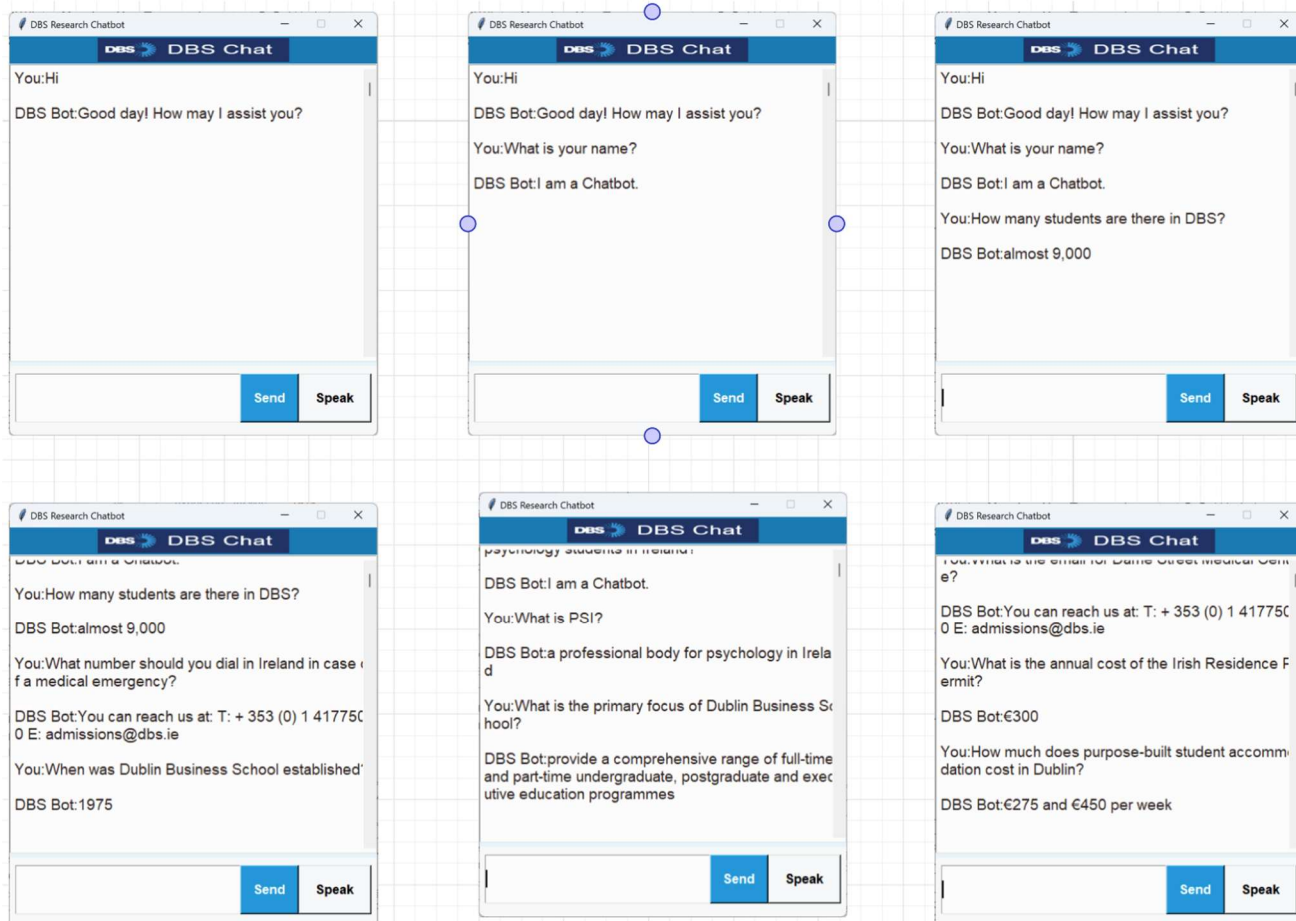


Figure 16 - Text mode conversation snapshots of user interface.

Sample Chatbot interaction - in voice mode through Speak button.

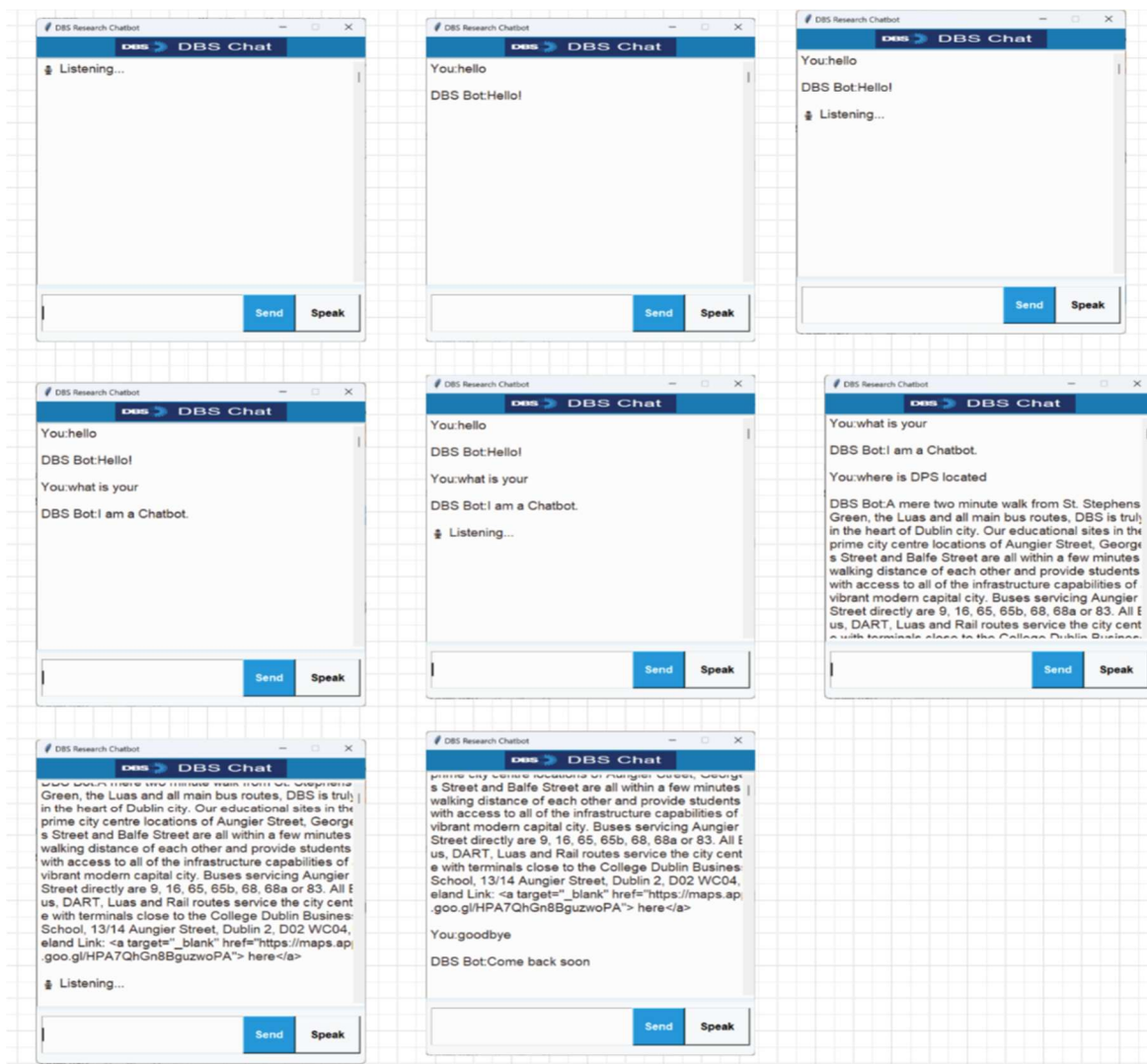


Figure 17 - Voice based conversation snapshot of the user interface.

Detailed steps of Model development

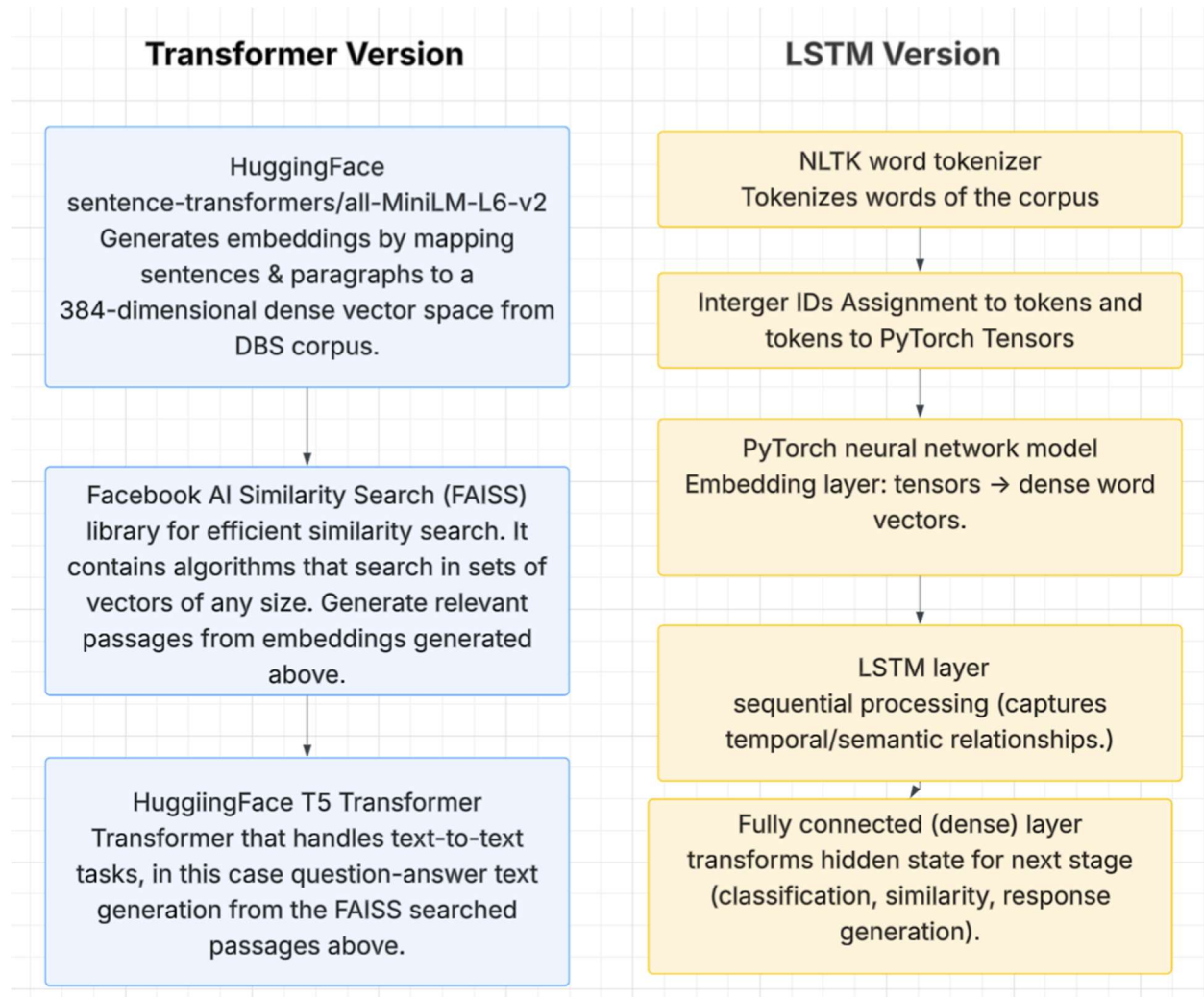


Figure 18 - Detailed steps of models' development.

9.0 Appendix B Artefacts and Repositories details

List of the project files and their purpose.

Table 1 - All project files explained.

Phase	Files	Purpose
Utilities and environmental setup files - Files used during data collection and cleaning process	install_dependencies.py	All the installation required are listed in it which helps in building up the environment before app is run
	doc_converter_1.py	Utility to convert the pdf into text form.
	dbswwebscraping.py	python script to web scrapes the contents from the website. In this case DBS web pages were being scraped.
	DBSWebScraping.ipynb	Collab version of the Web scraping file
LSTM & Transformer modeling - files used in data analysis and model building	LSTM_DBS_Chatbot_2025.py	Contains the code for the LSTM Version of the chatbot
	LSTM_DBS_Chatbot_2025.ipynb	
	Transformer_DBS_Chatbot_2025.py	Contains the code for the Transformer Version of the chatbot
	Transformer_DBS_Chatbot_2025.ipynb	
User interface and backend processing files- Integrating UI with backend	src\UI_Chatbot_Interface.py	user interface implementation of the chatbot.
	Model_Response_Predictor.py	Implement the backed processing by generating response to the user query provided by chat interface.
	src\Intent_Rule_Responses.py	detect intent and responds
	src\Voice_IO_Logic.py	Listen to the user when user hits the speak button on the chatbot
Result - files used for Models	Model_Comparison_Results_2025.pynb	Provide comparison of the LSTM and Transformer

benchmarking and Hypothesis testing		outputs and perform Hypothesis testing
	LSTM_DBS_Chatbot_2025_Results.docx	It contains the Average of evaluation metrics of LSTM variant
	LSTM_DBS_Chatbot_Output_2025.txt	It contains the evaluation metrics for each query in test for LSTM variant
	Transformer_DBS_Chatbot_2025_Results.docx	It contains the Average of evaluation metrics of Transformer variant
	Transformer_DBS_Chatbot_Output_2025.txt	It contains the evaluation metrics of all the queries of test data for Transformer variant
Data Files		
	Intent-Common.json	General intent and response pairs.
	Transformer_Training_DataSet-1.txt	Training dataset file-1 for Transformer variant
	Transformer_Training_DataSet-2.txt	Training dataset file-2 for Transformer variant
	Transformer_Test_DataSet.json	Testing dataset file for Transformer variant
	LSTM_Testing_DataSet.json	Training dataset file for LSTM variant
	LSTM_Training_DataSet.json	Testing dataset file for LSTM variant

Github link to the project - The full source code, datasets, and notebooks related to this research project are available at the GitHub Repository: [DBS MSc Applied Research Project 2025](#)

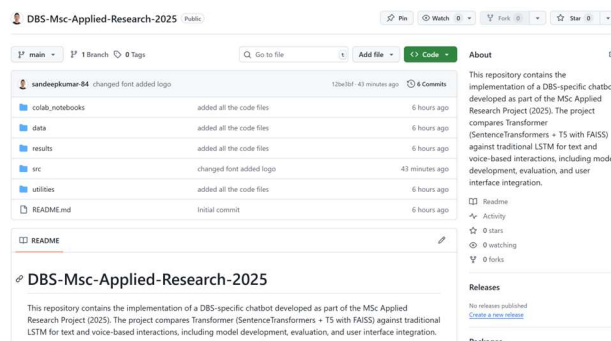


Figure 19 - GitHub repository snapshot.

Hugging face repository - link to the repository where trained transformer model is deployed is given below. This is to save retrain time when used during chatbot implementation or reused without retraining it with DBS data. Hugging Face Repository : <https://huggingface.co/sandeepkumar84/dbs-chatbot-transformer-hf-v3>

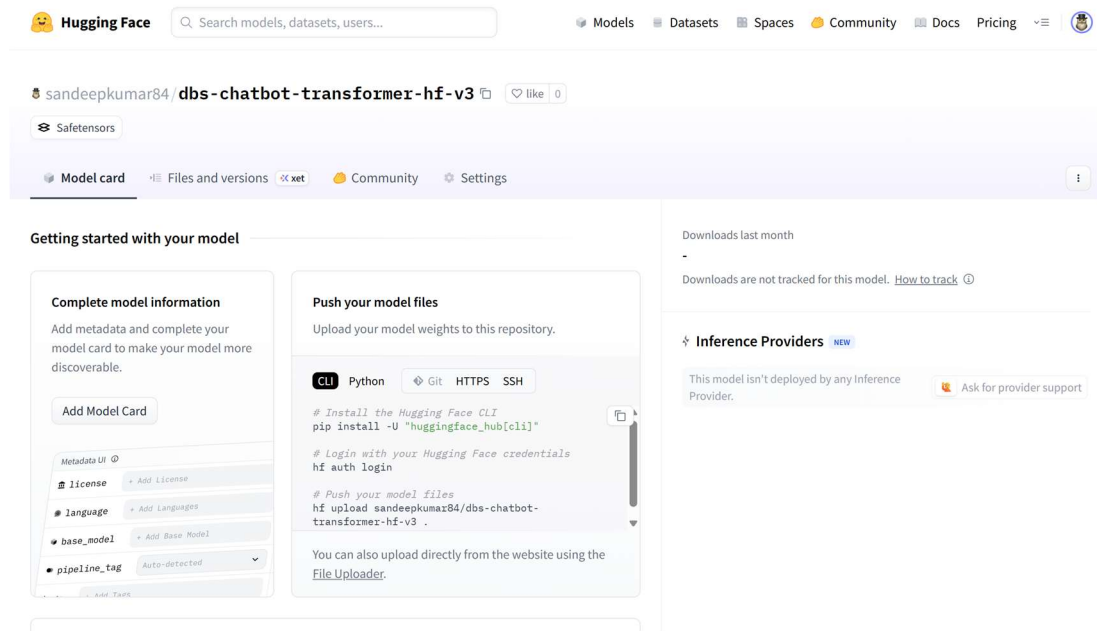


Figure 20 - Hugging face repository snapshot.

Tools & Libraries – Apart from standard ones, below tools and libraries are utilized in the project.

Table 2 - Software Libraries used in the project.

Library	Usage
Visual Studio Code	IDE for coding
Google Colab	Coding and testing
Tkinter	Generating UI
Python environment.	Overall coding
HuggingFace	Model deployment
Matplotlib	Core plotting library
WordCloud	For generating word cloud visualizations
pyttsx3	Text to Speech (TTS) library for Python
Docling	For PDF/Docx to text conversion
BeautifulSoup4	For web scraping and HTML parsing
Scikit-learn	ML tasks

Sentence-Transformers	For creating sentence embeddings
all-MiniLM-L6-v2	Hugging face sentence transformer
t5-base	Hugging face Text-To-Text Transfer
faiss	A library for efficient similarity search and clustering of dense vectors.
Transformers	For pretrained NLP model loading and generation
PyTorch	Core deep learning framework

Set-up and readme instructions

Option 1 – Manual - Copy the folder “**contents**” directly into the C drive. Move to the “src” directory where code files are present. Run installation file, it will install all the required libraries. python “utilities/install_dependencies.py”. To run the Transformer version of the chatbot use command python “src/Transformer_DBS_Chatbot_2025.py”. To run the LSTM version of the chatbot use command python “src/LSTM_DBS_Chatbot_2025.py”. To run the chatbot user interface application use command python “src/UI_Chatbot_Interface.py”. To run the result comparison and hypothesis testing results python “src/ResultComparison.py”

Option 2 – Auto - open the installation file below and update its variable like create_dir_and_upload_files_flag = True. Then run the file. It will first install the libraries, then create, necessary folders and finally upload files from GitHub python “utilities/install_dependencies.py“. Next move to the “src” directory where code files are present. repeat steps 10.5.1.4 through 10.5.1.7.

11.0 References

1. Turing, A.M. (1950) 'Computing Machinery and Intelligence', *Mind*, 59(236), pp. 433–460. Available at: <http://www.jstor.org/stable/2251299>
2. Weizenbaum, J., 1966. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), pp.36-45.
3. Colby, K., 1975. *Artificial Paranoia: A Computer Simulation of Paranoid Processes*. New York: Elsevier.
4. Shum, H.Y., He, X.D. and Li, D., 2018. From Eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19, pp.10-26.
5. Wallace, R.S., 2009. *The Elements of AIML Style*. ALICE A.I. Foundation. Available at: <http://www.alicebot.org>
6. Elov, B.B., Khamroeva, S.M. and Xusainova, Z.Y., 2023. The pipeline processing of NLP. In *E3S Web of Conferences* (Vol. 413, p. 03011). EDP Sciences.
7. Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
8. Pennington, J., Socher, R. and Manning, C.D., 2014, October. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).
9. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R., 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6), pp.391-407.

10. Žukauskas, P., Vveinhardt, J. and Andriukaitienė, R., 2018. Philosophy and paradigm of scientific research. *Management culture and corporate social responsibility*, 121(13), pp.506-518.
11. Saunders, M., Lewis, P. and Thornhill, A., 2009. *Research methods for business students*. Pearson education.
12. Mertens, D.M., 2017. Transformative research: Personal and societal. *International Journal for Transformative Research*, 4(1), pp.18-24.
13. Rossman, G.B. and Wilson, B.L., 1994. Numbers and words revisited: Being “shamelessly eclectic”. *Quality and quantity*, 28(3), pp.315-327.
14. Behl, V. and Bibhu, V. (2024) 'Leveraging Transformer Networks for Enhanced Conversational AI', in *2024 7th International Conference on Contemporary Computing and Informatics (IC3I)*, Greater Noida, India, pp. 937-943. doi: 10.1109/IC3I61595.2024.10829231. Available at: <https://ieeexplore.ieee.org/abstract/document/10829231>
15. Vaswani, A. et al., 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc. [online] Available at: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
16. Creswell, J.W. and Creswell, J.D., 2018. *Research design: Qualitative, quantitative, and mixed methods approaches*. 5th ed. Thousand Oaks, CA: Sage Publications.

17. Robson, C., 2002. Real world research.
18. Lin, J. and Michailidis, G. (2024) Deep Learning-based Approaches for State Space Models: A Selective Review. arXiv. Available at: <https://doi.org/10.48550/arXiv.2412.11211>
19. Schuurmans, D., 2023. *Memory Augmented Large Language Models are Computationally Universal*. arXiv. Available at: <https://doi.org/10.48550/arXiv.2301.04589>
20. Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K., 2018. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv preprint arXiv:1810.04805. Available at: <https://doi.org/10.48550/arXiv.1810.04805>
21. Hochreiter, S. & Schmidhuber, J., 1997. *Long short-term memory*. Neural Computation, 9(8), pp.1735–1780. doi:10.1162/neco.1997.9.8.1735.
22. Chung, J., Gulcehre, C., Cho, K. and Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555. Available at: <https://doi.org/10.48550/arXiv.1412.3555>
23. Daniel Jurafsky and James H. Martin. 2025. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edition. Online manuscript released January 12, 2025. <https://web.stanford.edu/~jurafsky/slp3>.

24. Glass, J., Flammia, G., Goodine, D., Phillips, M., Polifroni, J., Sakai, S., Seneff, S., Zue, V., 1995. Multilingual spoken-language understanding in the MIT Voyager system. *Speech Communication*, 17, pp.1-18.
25. A. Freed, *Conversational AI: Chatbots that work*, Manning, October 2021.
26. Patil, K., Patil, R., Koyande, V., Thakur, A.S. and Kadam, K., 2024, October. Analyzing Chatbot Architectures Utilising Deep Neural Networks. In 2024 IEEE 6th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA) (pp. 15-19). IEEE.
27. Tapia-Hoyos, J. J (2024). Chatbots in the service of university students: A review. *Social Sciences in Brief*, 1, 1-7. <https://doi.org/10.47909/ssb.08>.
28. Nguyen, T.T., Le, A.D., Hoang, H.T. and Nguyen, T., 2021. NEU-chatbot: Chatbot for admission of National Economics University. *Computers and Education: Artificial Intelligence*, 2, p.100036.
29. Modiba, Mashilo & Shekgola, Mahlatse. (2024). Utilising Artificial Intelligence Chatbots for Student Support at Comprehensive Open Distance E-learning Higher Learning Institutions in the Fifth Industrial Revolution. *Journal of Education, Society & Multiculturalism*. 5. 26-48. 10.2478/jesm-2024-0003.
30. Saygin, A. P., Cicekli, I., & Akman, V. (2000). Turing Test: 50 years later. *Minds and Machines*, 10(4), 463–518

31. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
32. Al-Amin, M., Ali, M.S., Salam, A., Khan, A., Ali, A., Ullah, A., Alam, M.N. and Chowdhury, S.K., 2024. History of generative Artificial Intelligence (AI) chatbots: past, present, and future development. *arXiv preprint arXiv:2402.05122*.
33. Cambria, E. and White, B., 2014. Jumping NLP curves: A review of natural language processing research. *IEEE Computational intelligence magazine*, 9(2), pp.48-57.
34. Manning, C.D., 2009. *An introduction to information retrieval*.
35. Liu, Y., Liu, Z., Chua, T.S. and Sun, M., 2015, February. Topical word embeddings. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 29, No. 1).
36. Young, T., Hazarika, D., Poria, S. and Cambria, E., 2018. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence magazine*, 13(3), pp.55-75.
37. Johnson, B. and Christensen, L., 2000. *Educational research: Quantitative and qualitative approaches*. Allyn & Bacon.
38. Bender, E.M., Hovy, D. and Schofield, A., 2020, July. Integrating ethics into the NLP curriculum. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts* (pp. 6-9).

39. European Union (2016) *Regulation (EU) 2016/679 of the European Parliament and of the Council*. Official Journal of the European Union. Available at: <https://eur-lex.europa.eu/eli/reg/2016/679/oj> [Accessed 17 Jul. 2025]

40. Artificial Intelligence Act, 2024. *Annex IV – Technical Documentation*. [online] Available at: <https://artificialintelligenceact.eu/annex/4/> [Accessed 17 Jul. 2025]

41. Papineni, K., Roukos, S., Ward, T. and Zhu, W.J., 2002, July. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics (pp. 311-318).

42. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q. and Artzi, Y., 2019. Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675.

43. Wilcoxon, F., 1992. Individual comparisons by ranking methods. In Breakthroughs in statistics: Methodology and distribution (pp. 196-202). New York, NY: Springer New York.