University of Kerbala
College of Computer Science & Information Technology
Computer Science Department

Chatbot for Improving Marketing Activities Using Deep Learning

A Thesis

Submitted to the Council of the College of Computer Science & Information Technology / University of Kerbala in Partial Fulfillment of the Requirements for the Master Degree in Computer Science

Written by

Alaa Tuama Abdulazeez Altuama

Supervised by

Asst. Prof. Dr. Dhamyaa A. Nasrawi

2023 A.D. 1444 A.H.

Supervisor Certification

I certify that the thesis entitled (**Chatbot for Improving Marketing Activities Using Deep Learning**) was prepared under my supervision at the department of Computer Science/College of Computer Science & Information Technology/ University of Kerbala as partial fulfillment of the requirements of the degree of Master in Computer Science.

Signature:

Supervisor Name: Asst. Prof. Dr. Dhamyaa A. Nasrawi

Date: / /2023

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "Chatbot for Improving Marketing Activities Using Deep Learning" for debate by the examination committee.

Signature:

Dr. Asst. Prof. Dr. Muhannad Kamil Abdulhameed

Head of Computer Science Department

Date:13 /7 /2023

Certification of the Examination Committee

We hereby certify that we have studied the dissertation entitled (**Chatbot for Improving Marketing Activities Using Deep Learning**) presented by the student (**Alaa Tuama Abdulazeez**) and examined him/her in its content and what is related to it, and that, in our opinion, it is adequate with (**Excellent**) standing as a thesis for the degree of Master in Computer Science.

Signature:
Name: Dr. Ayad Rodhan Abbas
Title: Prof.
Date: / / 2023
(Chairman)

Signature:
Name: Dr. Noor Dhia Al-Shakarchy
Title: Assistant Prof.
Date: / / 2023
(Member)

Signature:
Name: Dr. Ayad Hameed Mousa
Title: Assistant Prof.
Date: / / 2023
(Member)

Signature:
Name: Dr. Dhamyaa A. Nasrawi
Title: Assistant Prof.
Date: / / 2023
(Member and Supervisor)

Approved by the Dean of the College of Computer Science & Information Technology, University of Kerbala.

Signature:
Name: Dr. Ahmed Abdulhadi Ahmed
Title: Assistant Prof.
Date: 13 / 7 / 2023
(Dean of College of Computer Science & Information Technology)

Dedication

Lovingly dedicated to..

My respective family, for without their love and support, none of this would be possible.

My father, for his continuous words of wisdom, makes me take solid steps every day.

My mother continuously nurtured me with her love and devotion.

My brothers Safaa, Bahaa, and Alaa stand with me along the time and support me.

Alaa Tuama

Acknowledgement

In the name of Allah, the merciful and Most Gracious. Praise and thanks are due to Allah for his guidance, mercy, and blessing, for taking care of me every step of the way and helping me complete this thesis.

My heartfelt thanks to my supervisor Asst.Prof.Dr. Dhamyaa A.Nasrawi for her assistance and continuous support. I appreciate her encouragement, kindness, guidance, and contribution, which provided a good basis for the success of this thesis.

My sincere thanks go to Asst.Prof.Dr. Noor Dhia Al-Shakarchy for her great support. Her valuable advice and constructive comments have been of great value throughout the development of this thesis.

I'd like to extend my gratitude to the Computer Science Department staff and faculty, who worked tirelessly to provide us with relevant scientific content and support throughout my thesis. Additionally, I would like to thank the Dean and Deputy Deans of the College of computer Science and IT and the head of the computer science department Asst.Prof.Dr.Muhannad Kamil Abdulhameed for their support during this period.

My deepest gratitude goes to my family for their patience, support, and encouragement. I would like to say special thanks to my friends for their constant support.

Alaa Tuama

Abstract

With rapid technological advancement, more organizations move from manual customer support to automated processes. In addition, the demand for conversational agents has increased dramatically. Chatbots are conversational agents that simulate conversations with humans through natural language, including voice and text.

Chatbots are typically implemented through either generative or retrieval-based methods. Retrieval-based chatbots rely on predefined answers and lack of flexibility, while generative-based model generates new responses. In addition, customers require assistance when purchasing products which providing this service take a significant amount of time and effort. The main aim of this thesis is to improve a smart marketing chatbot (SMC) and user request understanding. To achieve these aims, two models was proposed; the first is a generative-based chatbot using the seq2seq LSTM model. The second model understands user requests using intent classification and named entity recognition. Models are based on question and answer Amazon office products.

The intent classification implemented using deep learning with the BiLSTM model. The traditional approach of Name Entity Recognition (NER) is insufficient to extract relevant information based on the particular domain of this thesis. To overcome this, custom NER based on BiLSTM employed to provide more accurate and contextualized extraction for the proposed chatbot conversation.

The results show that the chatbot handled user questions fittingly with BLEU score of 57.38. The intent classifier model achieved 94.75% accuracy. Finally, custom NER based on BiLSTM achieved 96.94% accuracy. Moreover, in a comparison of proposed models and relevant researches, the proposed models outperform others. The performance of all models has been improved compared to those in previous studies especially after preprocessing and annotating the dataset as well as customizing the named entities.

Declaration Associated with this Thesis

1. "A Survey on the Impact of Chatbots on Marketing Activities," A. T. Al-Tuama and D. A. Nasrawi, *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2022, pp. 1-7, doi: 10.1109/ICCCNT54827.2022.9984635

2. "Intent Classification Using Machine Learning Algorithms and Augmented Data," A. T. Al-Tuama and D. A. Nasrawi, *2022 International Conference on Data Science and Intelligent Computing (ICDSIC)*, Karbala, Iraq, 2022, pp. 234-239, doi: 10.1109/ICDSIC56987.2022.10075794

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
AIML	Artificial Intelligence Markup Language
ALICE	Artificial Linguistic Internet Computer Entity
ANN	Artificial Neural Network
API	Application Programming Interface
ASIN	Amazon Standard Identification Number
ATIS	Airline Travel Information System
BERT	Bidirectional Encoder Representations Transformers
BI	Begin-Inside
BiLSTM	Bidirectional Long Short-Term Memory
BIOS	Begin-In-Out-Single
BLEU	Bilingual Evaluation Understudy
BRNN	Bidirectional Recurrent Neural Network
CBR	Customer-Brand-Relationship
CME	Chatbot Marketing Efforts
CNN	Convolutional Neural Network
CRF	Conditional Random Field
CSR	Customer Service Representatives
CSV	Comma Separated Values
CUSM	Covenant University Shopping Mall
DNN	Deep Neural Networks

FAQ	Frequent Asked Question
GRU	Gated Recurrent Unit
IR	Information Retrieval
LMCL	Large Margin Cosine Loss
LOF	Local Outlier Factor
LR	Logistic Regression
LSTM	Long Short-Term Memory
NB	Naïve Bayes
NER	Named Entity Recognition
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
RNN	Recurrent Neural Network
Seq2Seq	Sequence to Sequence
SMC	Smart Marketing Chatbot
SVM	Support Vector Machine
TF-IDF	Term Frequency - Inverse Term Frequency

CHAPTER ONE

INTRODUCTION

1.1 Overview

Nowadays, the use of Artificial Intelligence (AI) is becoming increasingly integrated into daily life. One of the eras of artificial intelligence that are an exciting technology is intelligent agents. These agents are intelligent software and hardware used to improve the quality of our lives [1]. Chatbots are conversational agents or assistance AI that simulate conversations with humans through natural language, including voice and text.

The word chatbots is a derivative of “chat robots,” understood as machine agents that serve as natural language user interfaces to data and services through text or voice. Different names of chatbots: AI assistance, virtual agent, machine conversation system, and chatterbot [2].

Chatbots allow users to ask questions or make commands in their everyday language and get the needed content or service conversationally. If chatbots gain the expected widespread uptake, this technology will dramatically change how people interact with data and services online.

This intelligent system is used in various fields, including marketing, education, support systems, health care, entertainment, information retrieval, business, and e-commerce [1]. Chatbots are an essential part of the modern online experience and they have been integrated into many websites, including Facebook, Telegram, and Google. Chatbots help customers to find relevant information about products [3]. The number of brands and businesses that already use chatbots is increasing. The chatbot's responsibility will be a large part of future digital marketing tasks. The conceptualization of the chatbot is attributed to the Turing test [4].

Recently, chatbots have been developed and enhanced by natural language processing, machine learning, and deep learning to analyze user queries and provide relevant answers. The fast progress of these techniques in the last few years has made it possible to build chatbots with human-like conversation.

1.2 Problem Statement

Digital marketing is based on the Internet to promote brands and connect them with potential customers. However, customers require an assistant to help them purchase products, facilitate online shopping, and provide services. The 24/7 service are require a lot of time and effort. Retrieval-based limited to predefine answers and has no ability to produce new responses. In addition, understand the intention of user queries help in fast responding and increase the accuracy. Moreover, the standard named entity recognition does not applicable in all domains. Therefore, the chatbot can provide these services, improving marketing, increasing sales, and connecting product information with possible clients. Therefore, this thesis attempts to develop a smart marketing chatbot that meets most user requirements and provides customers with accurate and efficient information. This smart marketing chatbot uses AI and natural language processing to make the user experience faster, more flexible, more intuitive, as well as it can give responses back to users based on their requests.

1.3 The Aims of the Thesis

This thesis comprehends the following aims:

1. Improve a smart marketing chatbot as a generated-based approach to decrease the chance of response errors because the response occurs in real-time.
2. Understand the user's request including intents classification, and detect custom entities from user requests.

1.4 The Objectives of the Thesis

To accomplish the above aims, a new smart marketing chatbot system is proposed and experimentally evaluated in terms of effectiveness and efficient performance in detail:

1. Design a new architecture of smart marketing chatbot for a specific domain using deep learning model sequence to sequence long short-term memory LSTM as a generative-based approach to enhance the response to users' requests.
2. Annotate the dataset into labels and classify intents using deep learning bidirectional long short-term memory BiLSTM to understand the user's intentions.
3. Annotate the dataset with a custom-named entity, and then extract the entities from the office products dataset from Amazon to interpret what users ask.
4. Compute automatic-based metrics to evaluate them in term of effectiveness.

1.5 Related Works

This section introduces previous studies relevant to this thesis, such as essential criteria for developing chatbots, intent classification, and entity recognition chosen for the review. Reviewing relevant studies are divided into the following subsections.

1.5.1 Intent Classification

(Lin & Xu, 2020) proposed a method to detect unknown intent by using two methods. Firstly, train the known intents using the BiLSTM as a feature extractor. After that, learn the deep discriminative features through large margin cosine loss (LMCL) and the local outlier factor (LOF) used to extract unknown intents. The model decreased intra-class variance and increased inter-class variance. The study conducted on two datasets, ATIS and SNIPS. The results show that LOF (LMCL) outperforms other compared baseline methods. The f1 score of LOF model on ATIS dataset is between 39.6-84.1, SNIPS gained 79.1 for f1 score. The performance of the unknown intent detection on the ATIS dataset is less than SNIPS dataset because the known intents of ATIS are very similar, and close together, so the overlapping of unknown intent with the known intents is high. However, the known intents of SNIPS are distinguished from different domains and dissimilar from each other's [5].

(Schuurmans & Frasincar, 2020) They have explored several machine learning techniques for classifying the intents of dialogue systems and comparing these techniques to find the best performing method. They used the combination of a bag of words with Naïve Bayes

(NB), a continuous bag of words with Support Vector Machine (SVM), bidirectional long short-term memory (LSTM) networks, and hierarchical. The experiment was conducted on three corpora, Travel Scheduling corpus, Ask Ubuntu corpus, and Web Applications. The authors implemented the experiment on the complete dataset, which includes all datasets and individual datasets that consider each subset individually. The results show that flat SVM outperforms other methods in the complete datasets with 75.2 f1 score, and the hierarchical SVM performs best with 78.2 f1 score. In individual datasets, the macro-F1 score in the travel scheduling corpus BiLSTM Word2Vec with 98.2 , in asking Ubuntu the SVM fast-text average with 81.2 , in web applications also SVM fast-text average with 77.1 had the highest score and best performance [6].

(Zhang et al., 2021) Proposed an adaptive decision boundary (ADB) that extracts open domain intent and specifically known intents. The known intents are labeled, but the open domain has no labeling. This research based on three datasets named banking, OOS, stack overflow. The researchers first used BERT to extract intent features and pre-trained labeled intents. Initialized a centroid and specified known class decision boundaries. After that, researchers proposed a loss function to optimize the hyperparameters of boundaries. The range of accuracy results is between 78.85-86.32, and the f1-score is 71-62-85.99 [7].

(N et al., 2021) Developed a contextual goal-oriented chatbot that remembers the conversations with a specific user and provides the appropriate answer by extracting the intent from the conversations. The dataset used in this research based on networking domain. They used two models to find intent: The Bert and goBot models. The results show that the goBot gained 0.76 accuracies; overall, the models have good accuracy and can be improved on large data [8].

1.5.2 Named Entity Recognition

(Schweter & Akbik, 2020) They have performed a comparative evaluation of document-level NER features. They Perform their experiments on the CoNLL dataset using two different approaches. The first approach used a fine-tun transformer, and the second used a feature-based transformer to add features to LSTM-CRF architecture. Fine-tuning typically only adds one linear layer to a transformer and modifies the entire architecture. The transformer is used in feature-based approaches to create embedding for each word in sentences, which are then fed into LSTM-CRF architecture. The results show that fine-tuning outperforms the feature-based method with 97.02 f1 score. Overall, document level has a good impact on NER quality. Finally, they added FLERT as an extension to the FLAIR framework [9].

(Ali, 2020) the researcher of this paper focuses on named entity recognition (NER) and intent classification that was finally integrated into the chatbot. The NLU contains two components intent prediction and entity extraction. NLG is responsible for generating responses according to the knowledge base. ANN with two hidden layers is used for intent classification. ANN with two hidden layers and four units for the output layer is used to design the NER model and extract the entities from the sentences without their location. Entities was created if the message includes a specific entity rather than remaining empty. Entity types include names, locations, organizations, and miscellaneous. The experiment of NER was conducted on CoNLL-2003. The result of entity extraction was 81.66% for the f1-measure. Intent classification gains 89% accuracy [10].

(Straková et al., 2020) The researchers present two neural network models for nested named entity recognition (NER), where named entities may overlap and have multiple labels. The first model employs a regular LSTM-CRF predictor and combines the nested entity's labels into a single multi-label. This model has the dual benefits of being easy to implement and accurate; Large increases in NE class sizes are its drawback. In the second, nested NER considered as a sequence-to-sequence. The proposed approaches outperform nested NER on ACE-2004, ACE-2005, GENIA, and Czech CNEC. Furthermore, the results show that the seq2seq gained range between 78.31-86.88 of f1 score for each dataset[11].

1.5.3 Chatbot

(Cui et al., 2017) proposed a new AI assistance super-agent; this agent is a customer service chatbot that uses data from online shopping sites (Amazon) to answer questions. Super-agent can identify the type of chat by using crawlers to crawl the HTML data when the product page is first visited and then split them into many types of messages such as chit-chat, fact QA, FAQ search, and customer review, thus creating higher confidence score answer back as a response to the customer. Super-agent uses machine learning techniques and NLP. Super-agent is a chatbot that helps with FAQs. It obtains information from different e-commerce sites to answer frequently asked questions, thus freeing up the workload of support staff to answer much more detailed questions. Super-agent is an add-extension to the web browser, providing large-scale and publicly available data. However, the super-agent lacks intent query detection [12].

(Xu et al., 2017) create a customer service chatbot on social media that automatically generates customer responses. The system is adopted with deep learning techniques that can be implemented for seqtoseq learning with LSTM and word embedding. This bot for the training model used approximately 1M Twitter conversations between users and 60 different brands. The results show that there are two types of requests. The first one is the emotional request, which includes customers' emotions and opinions toward the brand, and it owns more than 40%. The second one is the informational request that allows users to ask questions to get information about a specific product. Lastly, the deep learning model and information retrieval (IR) results evaluated by three measures (appropriations, empathy, and helpfulness) show that the deep learning model has outperformed the information retrieval system. The drawback of this chatbot is that in the case of empathy ratings, the system's performance dropped when users switched from emotional to informational requests[13].

(Nursetyo et al., 2018) used AIML to develop an artificial intelligence chatbot for an online servicing customer in the telegram application. The corresponding chatbot was built using a knowledge base and implemented in an Indonesian restaurant. The user's input processed into three stages: data parsing, pattern matching, and AIML crawling data. The question was classified into three types (general questions, calculations, and stock checking). The test was conducted by asking 300 user questions to measure the chatbot system response time. The results have shown that the bot correctly responds to questions with an average response time of 3.4 seconds. In addition, the study had a 100% response accuracy by using correct formal words and sentences. However, the system needs to recognize users' misspelling mistakes [14].

(Muangkammuen et al., 2018) Proposed and developed a chatbot for e-commerce that automatically responds to FAQs by customers. The system used RNN in the form of LSTM for text classification. In this article, the researchers focused on developing a chatbot using machine learning based on retrieval rather than a generative model, spending little time developing the system. The user inputs Thai text questions and then processes them in the classification model to recognize the right group of questions. Next, the overall classes are categorized manually into 80 pairs of questions and answers. Finally, the system replies with appropriate answers according to existing classes. The result shows that the chatbot could process 86.36% of questions with 93.2% accuracy of correct answers [15].

(Arsenijevic & Jovic, 2019) They described in their study the role of artificial intelligence chatbots in marketing. This study aims to investigate the communication of respondents with virtual agents of organizations, Respondent's behavior, expectations, and habits when using different communication channels, especially Internet communication channels (chatbots) in their daily work. The results show that using chatbots in the marketing service provides simple, fast information. Therefore, the organization should consider the benefits of chatbots in their communication with customers [16].

(Wibowo et al., 2020) Explore the use of chatbots in the customer service area for an e-commerce website and how they can improve to achieve higher user satisfaction. First, researchers create a database that stores basic product information, such as price, color, and quantity, and will use it to generate answers. Then, the chatbot is adapted via deep learning and NLP, where the user inputs the query input. The bot searches for a keyword in the sentence and performs pattern matching to answer questions and provide details that help the customer. The bot can be easily accessed through the e-commerce website or mobile application, providing precise information and handling simple user problems. Lastly, they use a questionnaire to investigate how people feel about chatbots. The results showed that most users preferred using a chatbot in marketing and found it helpful. In addition, consumers think that the chatbot provides better shopping services and increases online purchasing [17].

(Khan, 2020) developed an e-commerce sales chatbot that helps improve customer relationships and increase sales. The main objective of this project is to create a more accurate modular chatbot architecture and add new features that are more accessible. The microservice architecture was chosen to create the system. The system is composed of five components. The first component is the NLU Engine, an essential part of the model. The second is the Recommendation Engine, which suggests products users might want to buy. The adaptive Pricing Engine collects product information from the recommendation engine and generates real-time customer discounts or deals. The system's last component and core is the Bot Engine, which uses the NLU engine to classify user input and then routes the user request to the specific controller. The controller creates a response for the user [18].

(Tran & Luong, 2020) has developed a chatbot to comprehend user utterances in the Vietnamese language better. To this end, the researchers use two neural architectures, BiLSTM and CNN for intent identifying and sequence labeling used to extract relevant contexts. The results show that they have achieved an F-measure of 82.32% in detecting intents while achieving results ranging from 78% to 91% regarding context extraction [19].

The authors (Manyu Dhyani and Rajiv Kumar, 2020) created an intelligent chatbot using a bidirectional recurrent neural network (BRNN) with an attention layer based on the Reddit dataset. For implementing this experiment, the Reddit dataset was converted to a database with five specific fields: id, parent comment, score, sub comment, and UNIX. This research aims to improve the model's BLEU score, learning rate and perplexity(probability distribution). The results after training are 30.16 for the BLEU score, 56.10 for perplexity and 0.0001 for the learning rate [20].

(Solis-Quispe et al., 2021) Proposed a chatbot to simplify customer interaction and e-commerce, making online retail purchases easier for users. This chatbot will result in a quicker purchase process and fewer interactions. For building the Chatbot, Dialogflow is used, and a database that includes product information is created. The web page (chatbot interface) is integrated with the Dialogflow platform. Chatbot also features the current FAQ and guides customers through the purchase process. It also recommends improvements to reduce customer frustration (e.g., recommending a closer store according to their location). The optimized purchase process reduces time by 32% and interactions by 43%, exceeding the expected value of previously proposed metrics [21].

(Oguntosin & Olomo, 2021) They proposed a web-based chatbot (Hebron) for Covenant University Shopping Mall (CUSM). The shopping mall had no online inventory service to allow students who patronize the mall to be disappointed that they could not check the stock availability of the items they wanted to buy online before visiting in person to shop for them. The Hebron chatbot system is divided into two parts (front-end, which includes the chatbot interface, backend consisting of machine learning and database). The result showed that the Hebron chatbot gains the user's satisfaction and alleviates the discomfort of covenant university members. On the other hand, the number of participants was limited [22].

(Naing Naing Khin, Khin Mar Soe, 2021) Builds a chatbot based on RNN seq2seq with an attention mechanism for educational purposes. This work aims to create a corpus for the university based on the Myanmar language and enhance the sequence to sequence model for the chatbot model. Chatbot provides related information and answers FAQs about the university. The dataset is in the Myanmar language, collected manually from students, and contains 5000 questions and answers. The attention layer solves the problem of long sequences by focusing on the source's content and establishing direct shortcut connections between the target and source. The results show that the model with an attention layer is better than a regular RNN. Also, the model is evaluated by BLEU score, in which RNN with attention mechanism gained 0.41 for BLEU score and regular RNN BLEU score is 0.35 [23].

(Bachtiar et al., 2022) Came up with two seq2seq models based on LSTM and GRU to make an open-domain generative-based chatbot. They took two datasets from Kaggle and merged them into one. The paper aimed to measure the BLEU score and model loss to compare the performance of the two models. The results show that the LSTM model outperforms the GRU model regarding answers, loss, and BLEU score, by gaining a train-loss of 0.7064, Val-loss of 8.9740, and BLEU score of 0.0588. On the other hand, GRU got 2.1125 for train loss, 7.1840 for Val loss, and 0.0028 for BLEU score. Both models need some modifications to generate better responses [24].

Table 1.1 presents the overview of the literature review.

Table 1.1: Overview of Literature Review

N	Ref	Dataset	Method	Result	Strength	Weakness
1	[5]	ATIS-SNIPS	Used BiLSTM	79.2-84.1 (f1-score)	Decrease intra-class variance, increased inter-class	Performance in ATIS low because of the high similarity of intents
2	[6]	Travel scheduling, Ask ubunto, web app	Used SVM, BiLSTM, Naïve base, hierarchical	75.2 flat, 78.2 hierarchical, 98.9 TR, 81.2 Ask, 77.1 web (f1-score)	High F1-measure results	Local hierarchical classifier

3	[7]	Banking, OOS, stack overflow	Used ADB and BERT	71.62-85.99 (f1-score)	Open intents classification without any prior knowledge	The model is more robust with less labeled data
4	[8]	Networking domain	Used BERT and gobot models	76.0 (Accuracy)	A good model for contextual chatbot	Small data
5	[9]	Conll	Used LSTM-CRF	97.02 fine-tun, 96.53 feature base (f1-score)	Added to the FLAIR framework	The model performed well on the English dataset rather than in other languages, specifically German.
6	[10]	Conll-2003	Used ANN	81.66 entity, 89.0 intent (f1-score)	The proposed method outperforms other compared methods.	No entity location
7	[11]	Ace 04-05, GENIA, Czech CNEC, conll 03	Seq2seq, LSTM-CRF	78.31-86.88 nested NER, 93.38 flat NER (f1-score)	The model outperforms other models.	Significant increases NE class sizes.
8	[12]	from Amazon	Used NLP and machine learning techniques.	-	Large-scale and publicly available e-commerce data	Weak Intent query detection
9	[13]	Twitter conversation	Used LSTM to generate responses	0.36 (BLEU score)	Generate responses automatically and high learning writing style from the brand	Low empathy rating
10	[14]	Indonesian restaurant	Used AIML and parsing data and pattern matching	-	High accuracy answers user questions	Does not recognize user misspelling mistakes
11	[15]	Thai-FAQ	Used RNN in the form of LSTM	93.2 (Accuracy)	Have a good FAQ corpus.	Small dataset
12	[16]	-	Study the effect of a chatbot on marketing and use a survey to collect data.	-	Provide simple, fast-obtained information	Restricted to university students only

13	[17]	Manually created	Used deep learning and NLP.	-	Cost-efficient, publicly available,	System based on keywords and matching with database.
14	[18]	Manually created	Used machine learning and NLP	-	Modular chatbot architecture, Availability on more platforms	Small dataset
15	[19]	Manually created	Used BiLSTM for intent detection and sequence labeling for extracting context	82.32% for intent , 78.91% for context extraction (f1 score)	High intent classification and context extraction	imbalanced dataset problem
16	[20]	Reddit dataset	BRNN	0.30% (Bleu score)	Used attention mechanism to process sentence length of (20-40 words), good dataset	The answers are inaccurate.
17	[21]	Manually created	Used Dialog flow platform	-	facilitate the purchase procedure and helps the customers in decision making	Using chatbot platforms
18	[22]	Manually created	Used deep learning and NLP	-	Provide an accessible, smart, and comfortable shopping experience.	A small number of a participant in the questionnaire
19	[23]	Collected from yangon university	Seq2seq	0.41 % (Bleu score)	RNN with the attention mechanism	Small dataset
20	[24]	Simple dialoge and daily dialog	Used seq2seq (LSTM-GRU) models	0.058 % (Bleu score)	Used a good number of dataset samples	Used a small number of epochs (15 epochs)

1.6 Thesis Organization

The thesis divided into five chapters. The summaries of the chapters are as follows:

Chapter One: Introduction and related work.

Chapter Two: This chapter details the theory used in this thesis.

Chapter Three: This chapter focused on the proposed method.

Chapter Four: In this chapter, the results are obtained and discussed.

Chapter Five: This chapter involves conclusions and suggestions for future works.

CHAPTER TWO

THEORETICAL BACKGROUND

2.1 Overview

This chapter presents the details of chatbot techniques and the theoretical background for this work, including an overview of the chatbot, a brief timeline of chatbots, components, benefits of customer service, types, and the role of natural language processing in the chatbot.

The dataset used in this work and some concepts we needed word embedding, data augmentation, and cosine similarity included in this chapter.

Intents classification and name entity recognition was demonstrated to understand user requests, as well as machine learning and deep learning techniques used in our work.

Finally, present the evaluation measures used to evaluate the performance of the proposed system.

2.2 Chatbot

As technology advances, more and more organizations are moving from manual customer support to automated processes. The increasing popularity of artificial intelligence is driving this trend (AI) approaches, such as chatbots. Chatbots are a vital part of providing automatic customer support that is efficient and effective [25].

Chatbots are conversational agents or assistance AI that simulate conversations with humans through natural language, including voice and text. They can mimic human conversation and entertain users [1][26]. Different names refer to chatbots, such as AI assistance, virtual agent, machine conversation system, and chatterbot [2]. The use of chatbots has evolved rapidly in recent years. Chatbots are currently being used in various domains like science [27], education [28][29], health care [30][31], marketing [32], [33], information retrieval [34], entertainment [35], e-commerce [36], business [37][38], and support systems[39][1][40]. In addition, these conversational agents can improve human interaction with machines [40].

Chatbots help users to find information about products and services on e-commerce sites. This intelligent system uses natural language processing and human-computer interaction to make the user experience faster and more intuitive[3]. Chatbot functions based on user commands, which generate a collection of sentences with matching answers and questions. These queries are matched or generated for relevance to the user input, providing the necessary information [41].

2.3 Brief Timeline of Chatbots

Chatbots are an essential part of the modern online experience. They have been integrated into many websites, including Facebook, Telegram, and Google. However, where do chatbots come from? The initial idea comes from the Turing test. The Turing Test was proposed in 1950 by Alan Turing's paper "Computing Machinery and Intelligence.” In the paper, he proposed a test where an interrogator had to determine which player was a human and which machine through a series of written questions [4]. The brief timeline of chatbot technology is presented in Table 2.1.

Table 2.1: Brief Timeline of Chatbots

N	Chatbot	Year of Release	Type	Description
1	ELIZA [4]	1966	Rule-based	First psychologist chatbot.
2	PARRY [42]	1972	Rule-based	Modeled the behavior of paranoid schizophrenic.
3	Jabberwocky [42]	1981	Rule-based	Simulate natural human chat in an interesting, entertaining, and humorous manner.
4	Tomy [42]	1985	Rule-based	This bot is a wireless robot that repeats any message recorded on its tape.
5	Loebner Prize [4]	1990	Rule-based	It took a standard Turing test format but with judges awarding the most human-like computer program.
6	Dr sbatiso [4]	1991	Rule-based	This bot is an automatic psychologist chatbot speech synthesis development.
7	ALICE [4]	1995	Retrieval-based	The ELIZA program inspired the program and can respond with many different answers based on the sentence it receives from a person.
8	Elbot [4]	2000	Retrieval-based	Elbot is a cheeky chatbot uses wit and sarcasm, a healthy dose of irony to entertain humans.

9	SmarterChild [4]	2001	Retrieval-based	This conversational bot offered personalized conversation and was considered a precursor to Apple’s Siri and Samsung’s S Voice.
10	Mitsuku [4]	2005	Retrieval-based + Generative-based	This bot can reason with specific objects. In addition, she can play games and do magic tricks.
11	IBM Watson [42]	2006	Retrieval-based	The computer was created to compete on the game show "Jeopardy!" In its first attempt, Watson could answer the questions right about 15 percent, but later it could beat human contestants regularly.
12	Siri [42]	2010	Retrieval-based + Generative-based	In 2011, Siri was introduced as an app on the iPhone 4S. Nuance Communications provide the application’s speech recognition engine.
13	Google Now [4]	2012	Retrieval-based + Generative-based	Google Now was created by Google to be used for Google Search Mobile App.
14	Alexa [4]	2015	Rule-based + Retrieval-based + Generative-based	Amazon created Alexa.
15	Cortana [4]	2015	Retrieval-based	Cortana is an intelligent assistant created by Microsoft.
16	Messenger Bots [42]	2016	Retrieval-based + Generative-based	This chatbot can provide business-to-consumer and business-to-business interaction through automated features and functions.

17	Tay [4]	2016	Rule-based + Generative-based	Microsoft introduced Tay, a chatbot designed to mimic the speech and habits of a teenage American girl.
18	Woebot [42]	2017	Retrieval-based + Generative-based	Woebot is an automated conversational agent that helps people monitor moods and learn about themselves.

2.4 Components of Chatbot

Before starting to build a chatbot, we should understand a couple of terms carefully to facilitate diving deep into designing a chatbot.
Here are some essential components to know about chatbots:

2.4.1 Intent

Intents are the intention of the user from the utterance. The purpose of intents is to define the user's intent, and each user statement must be matched to a predetermined intent. The difficulty is that each intent is a gate the dialogue must pass through to reach the dialogue. For example, a customer's intent can be a query, problem, request, issue, and so on [43]. If the bot cannot understand a user, the bot will not be helpful, no matter how hard work on other parts like managing conversations[44].
For instance: when a user says, “I want to order a pizza” or “Can I order fried fries,” the chatbot can understand that the intent is “order food"[42].
Intents are divided into two parts:

2.4.1.1 Casual intent

Casual intentions or Small Talk intents refer to conversation openers and closers such as Hello, good morning, good evening, thank you, etc. These intents typically elicit responses from the bot, such as "Hello!" What can I do to help? Alternatively, "It was a pleasure speaking with you," "Goodbye," or "We appreciate your reaching out to us." Training bots to comprehend these small-talk intentions enables businesses to manage all small-talk communications instead of simply removing the context from the interaction [45].

2.4.1.2 Business intent

On the other hand, business intentions are the intents that pertain to the bot's business operations. For instance, if it is a movie bot, a message from a consumer such as "When is Avengers: Endgame coming out?". It is a business objective, as the client wants to know the release date of Avengers: Endgame. The bot should be able to swiftly retrieve this information from the database using specified labels [45].

2.4.2 Entities

Entities are the metadata that is used within the context of the intents. In other words, helpful information like the place, time, and person. These entities help describe the intent and can be used by other services to provide custom responses to an intent.

An intent can have multiple entities as well. For example, an entity represents a count, volume, or quantity [42].

For instance: in the example mentioned above, in the intent "I want to order a pizza," the entity is pizza.

Another example about multiple entities:” Order me a jacket of size 38."

Composite entities:

Category: jacket

Size: 38 [42].

2.4.3 Utterances

Utterances are a different form of the same intent the user says to the bot in conversation.

For instances:

“I want to order pizza” “Can I have a pizza “.

These two utterances are different but have the same meaning of intent [42].

2.4.4 Responses

A conversational AI's purpose is to provide an acceptable response to a user's statement. A response is information the assistant returns to the user [46].

2.5 Benefits of Chatbots on Customer Service

Conversational chatbots are a new type of customer service; their main benefit is fast and accessible help. All studies reported that the rapid response of chatbots was accentuated [47]. In today’s world, many businesses rely on bots to do their work. In addition, bots help save money and provide other benefits for the business [42]. Here are some benefits of chatbots in marketing technology:

2.5.1 Provide Customer Support

Chatbots, such as Customer Service Representatives (CSR), are commonly used to provide support. To provide globally accessible customer support, chatbots can be programmed to answer user questions, even during holidays. Chatbots assist with user-generated query flows outside standard business hours and can reduce overall payroll costs. Instead of paying for one or more staff to work night shifts, the organization can rely on one chatbot to interact with multiple clients at once [48].

2.5.2 Business Revenue

Chatbots are a great way to attract clients. They ask relevant questions, and some clients are forwarded to the sales team or scheduled for discussion. These agents can sometimes help with sales conversion, depending on the nature of the business [49]. Chatbots have proven to be an effective way to provide customer service. Companies with chatbot support or that create a new chatbot are more successful in the market than those without one [42].

2.5.3 24/7 Availability

Providing 24/7 support to customers is very important to improve the customer experience. Nowadays, users can immediately reach an operator and get their questions answered [49].

2.5.4 Fast Response

Artificial intelligence (AI) can handle many customer interactions simultaneously and promptly respond to their inquiries. It improves customer engagement, which leads to more sales and saves time [49].

2.5.5 Accessibility

Chatbots provide an easier way to interact with customers. For example, customers can easily open websites or phone applications and start talking with the chatbot [42].

2.5.6 Insights

The sales representative might have difficulty remembering customer behavior, but bots can use machine learning and data science techniques to learn about consumer behavior [42].

2.6 Types of Chatbots

Chatbots are becoming increasingly popular as tools for customer service. Depending on the chatbot's specific domain, task, and goals, they can be broken down into several distinct categories based on their input and output capabilities. Different types of questions can be categorized to understand what kind of response a given chatbot can provide [50]. See Figure 2.1.

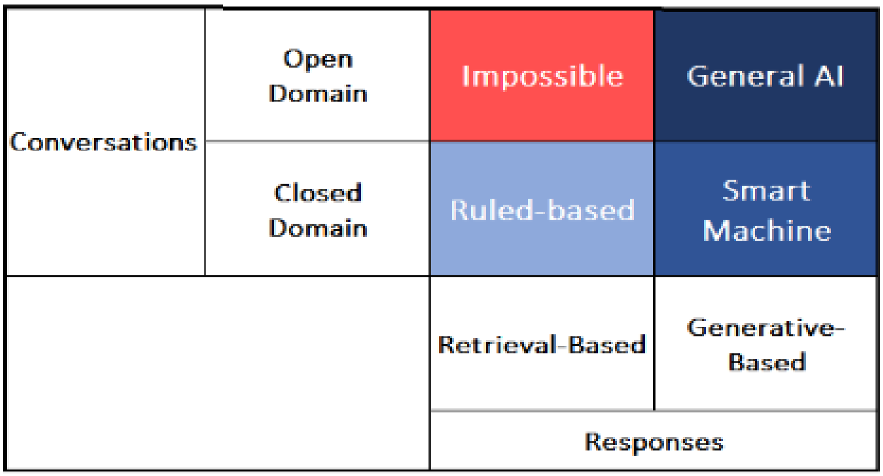

Figure 2.1: Types of Chatbot [51]

2.6.1 Knowledge Domain

In Chatbot development, knowledge domains are divided into two types– open and closed. Open-domain chatbots are designed to handle general inquiries and respond accordingly, without a specified goal or intention; this can be a challenging task due to their wide range of topics and facts. On the other hand, closed-domain chatbots are associated with one specific sector, providing specialized answers but limiting their overall capabilities. Both have value in delivering user requirements at varying difficulty levels when developing suitable Chatbot systems for business needs [50] [51].

2.6.2 Input and Response

The generated input and answer determine the other type of chatbot. This chatbot accepts natural language input, processes it, and generates natural language output. This form of chatbot is considered hybrid since it uses both natural language processing and rules to respond to user input. There are essentially two methods for eliciting a response. Answers can be derived from either a Retrieval-Based or a Generative-Based System [50].

2.6.3 Retrieved-based Model

Retrieval-based models are based on a repository of predefined responses. This model doesn't generate any responses only selects the best response based on the input and context [52][53]. End users follow a scripted conversation flow. Most chatbots are scripted, which means they have limited flexibility. Chatbots cannot handle unseen cases and will likely break if they do not have a script to follow. This chatbot generally deals with one domain with less flexibility and more rigid rules. Retrieval-based models do not generate grammatical mistakes [52].

2.6.4 Generative-based Model

The generative-based model can build new responses from scratch without relying on predefined responses. Generative model based on machine-translation techniques like seq2seq. This model's probability of making grammatical mistakes is high [53][54]. In terms of conversational chatbots, generative models are the superior option.

2.7 Role of Natural Language Processing in Chatbot

Natural Language Processing, or NLP as it is more commonly known, is a branch of artificial intelligence (AI) that looks at how computers can understand written and similarly spoken language to humans. In the case of bots, it helps determine what the users are trying to say and then creates responses based on contextual analysis, just like a person would [45]. NLP is critical for chatbots to understand and take action based on the user's input [55][56]. When a user says "hi," NLP understands the user means greeting, and upon that, it replies greeting to the user.

Without NLP, a chatbot cannot understand the meaning of each intent and differentiate between different intents like "greet" and "goodbye" It means that all the intents for a chatbot be just text-based users. Therefore, natural language processing extracts the message from insignificant text to meaningful text that a chatbot understands and recognizes [57].

2.7.1 Natural Language Understanding NLU

The assistant uses an NLU engine to determine what the user means when they say something. Natural language understanding (NLU) employs natural language processing and machine learning techniques to extract the user's intent and related entities from textual information of the user's input [58]–[61]. The NLU part is made up of two parts:

- Intent classifier: Finds the meaning that fits the whole sentence. The meaning is usually shown as a (verb-based) intent, and machine learning is used to pull it out.

- Entity extractor: Finds one or more subsets of the utterance that have a specific meaning, such as dates, places, and other objects. Most of the time, the entities are nouns, which can be pulled out by machine learning, hard-coded rules, or both.

2.7.2 Natural Language Generating NLG

While NLP transforms human language into unstructured data and NLU interprets data based on grammar, NLG produces text from structured data. The NLG is comparable to a data analyst who evaluates and turns information into words, phrases, and paragraphs. The NLG develops a contextualized story by encoding and expressing the meaning concealed within the data. After receiving feedback, the user can choose whether they have adequate information and decide whether to end the interaction or ask additional questions. This component's key advantage is that it provides a contextualized story by encoding and transmitting the hidden meaning in the data. The response generation component gets the intent and context information from the user message component. It uses Natural Language Generation (NLG) to respond to the user who appears as the person who wrote it. It used one of these three models to give the correct answers: the rule-based, retrieval-based, and generative models [1][62][63].

2.8 Amazon Office Products Dataset

The proposed dataset is question and answer office products that are gathered from Amazon product pages and are available on Amazon [64]–[66]. This dataset includes 33,984 questions with multiple answers (130,088 answers) in the office product domain. In addition, the dataset contains an Amazon Standard Identification Number (ASIN) number that is the product ID, question and answer type (yes/no) or open-ended, answer time is a raw timestamp, and Unix-time is the answer timestamp converted to Unix-time, question and answer text. An example of the dataset is presented in Figure 2.2 below:


```
{"Asin": "B000050B6Z", "Question Type": "yes/no", "Answer
Type": "Y", "Answer Time": "Aug 8, 2014", "Unix-Time":
1407481200, "Question": "Can you use this unit with GEL
shaving cans?", "Answer": "Yes. If the can fit in the
machine it will dispense hot gel lather. I've been using
my machine for both, gel and traditional lather for over
10 years."}
```


Figure 2.2: Example of Amazon Office Products Dataset [66]

2.9 Word Embedding

The mapping of words to vectors is known as word embedding. The features derived from this vector representation can be incorporated directly into an algorithm for machine learning. This process can be done in several methods, from using a primary count vector to employing deep learning strategies like Word2vec, One Hot encoding, TF-IDF, fasttext, and Glove [67]–[69].

2.9.1 One Hot Encoding

One Hot Encoding is a numerical data representation technique frequently used in machine learning applications. It involves mapping unique words to individual vectors of 0's and 1's, where the vector at each index corresponds to a given word equal to 1, while all other words in the vector are equal to 0 [70].

The one-hot vector of j, computed by equation 1:

$$x(j) = \begin{cases} 1 & if\ l = j \\ 0 & otherwise \end{cases} \quad (1)$$

Where: j ∈ $\{1, \dots, k\}$, k is length of vector. [71]

2.9.2 TF-IDF

TF-IDF is a method used to measure the relative importance of words. This method is done by multiplying Term Frequency (how often a word appears in a document) and Inverse Document Frequency (the frequency of the word in the corpus). In other words, TF-IDF looks at how frequently a term occurs in the text and how important it is when compared with the entire corpus [72]. The score of any word in any document can be represented as per in equation 2:

$$\mathrm{TFIDF(word, doc)} = \mathrm{TF(word, doc)} * \mathrm{IDF(word)} \quad (2)$$

The formulae to calculate TF and IDF explained in equations 3,4 respectively [73]:

$$\text{TF(word, doc)} = \frac{\text{Frequency of word} \in \text{the doc}}{\text{No.of words} \in \text{the doc}} \tag{3}$$

$$\text{IDF(word)} = \log_e\left(1 + \frac{\text{No.of doc}}{\text{No.of docs with word}}\right) \tag{4}$$

2.9.3 GloVe

Global Vectors for Word Representations (GloVe) is a highly acclaimed approach for creating word embedding. This count-based model learns words' semantic similarity by evaluating the underlying statistics of the corpus - such as words co-occurrence. In GloVe, global co-occurrence statistics are also taken into account. An additional benefit to this model is that it can be easily parallelized, allowing datasets with large volumes of data to be trained on [74][75]. Also, glove effectively studies conversational agents [76].

For some details, the matrix of word-word co-occurrence counts be denoted by X, whose entries X_{ij} tabulate the number of times word j occurs in the context of word i. The number of times any word appears in the context of word i computed by equation 5.

$$X_i = \sum_k X_{ik} \tag{5}$$

While the probability that word j appears in the context of word i computed by equation 6 [74].

$$P_{ij} = P(j|i) = {X_{ij}}/{X_i} \tag{6}$$

2.9.4 FastText

FastText is an advanced static word embedding technique that addresses the issue of earlier methods' neglect of a word's morphology. Unlike GloVe and Word2Vec, FastText considers a word's character n-grams to form its final vector, potentially giving it more accuracy for rare words. Additionally, this allows the FastText model to handle out-of-vocabulary (OOV) words as long as the respective character n-grams were seen during training time. This approach makes it a standout among other approaches used for static quality word embedding [75][77].

FastText method based on a continuous skip-grams, that each word represented as character n-gram. As a result, when training word embeddings, it can provide the vector for unseen words. The scoring functions to preserve the sub-word information computed by equation 7 [78].

$$s(w, c) = \sum_{n \in N_w} Z_g^T V_c \tag{7}$$

Where:

- w is word,
- c is vector,
- N_w is set of n-gram in w,
- N is dictionary size of n-gram,
- Z_g^T is vector representation assigned for each n-gram.

2.10 Data Augmentation

Data augmentation is a technique used to synthesize new data from existing information. It has many applications in Machine Learning, mainly in cases where the dataset is too small or imbalanced. Data augmentation helps generate more data representative of the original dataset. This enhances learning models by making them more robust and accurate [79]. Data augmentation in text poses an array of challenges. For instance, alterations such as reordering words can completely alter the intended meaning of a sentence, thus creating difficulties for proper implementation. Data augmentation by increasing the amount and diversity of training data can help models learn the underlying classes better. This can lead to improved performance on unseen data [80]. The lack of data is a common issue when implementing machine learning methods. Data augmentation is a solution that can be used to increase the amount of training data. This can be done by somehow transforming the available data [81]. There is some different way to augment data, such as substituting some words with their synonyms. Replace a few words by choosing words with close word embedding compared to those words (such as word2vec or GloVe). Exchange terms according to the context using strong transformer models (BERT). Employ back translation, which means translating a sentence from one language to another and then translating it back to its original language – this can sometimes alter several words.

Nlpaug library provides contextual word embedding Bidirectional encoder representations from transformers (BERT) that increase the training set by generating new utterances according to the context of sentences. This augmentation can help improve the number of utterances for each class, making class distributions more balanced and allowing for a more robust classifier [82]. There is a direct correlation between the amount of data used to train a model and that model's performance. This is seen in tasks of intent classification and question answering in natural language processing. Typically, the model is better trained with more data. However, acquiring and labeling this data can take time and effort [72][82][83]. Therefore, data augmentation is used for generating new text.

2.11 Cosine Similarity

Cosine similarity is a metric used to determine the degree of similarity between two vectors in an inner product space. Calculated by taking the cosine of the angle formed by two vectors, it is often employed in text analysis applications, where it can measure the similarity between documents. Various attributes can represent a document, including the frequency of certain words or phrases. These frequencies are recorded in a term-frequency vector, an object that represents each document. By analyzing the attributes stored in these vectors, one can gain insight into the contents of any given document [83]. The similarity equation is defined below:

$$sim(x, y) = \frac{x.y}{||x||||y||} \tag{8}$$

where $||x||$, $||y||$ are the Euclidean norm of vector $x = (x_1, x_2,..., x_p)$, $y= (y_1, y_2,..., y_p)$ respectively. A cosine value of 0 means that the two vectors have no match, while the smaller angle means the greater match between vectors [83].

2.12 Intent Classification

A vital component of a chatbot is the ability to classify user utterances into corresponding intents [84]. The classification of intents is a significant research area in artificial intelligence, with applications in areas such as product design and marketing to intelligent communication [72]. Intent classification classifies user input text into predefined intent categories based on the domains and intents involved. Intent detection is crucial in dialogue systems, allowing the system to understand users' wants. Intent detection is essential in systems where multiple users with different goals interact with the system simultaneously [85]. In addition, classifying intents are essential for businesses because it allows them to be more customer-centric. This is especially true in customer support and sales areas, where faster response times and personalized service can make a big difference [86]. There are several reasons why market intelligence is so crucial for businesses, including understanding and managing branding, knowing what competitors are up to, retaining customers, and figuring out the best fit for products and services in the marketplace [87]. However, extracting intent from the dialogue system is challenging [7]. The purpose of intent classification is to determine what the customer is trying to achieve [6].

2.13 Named Entity Recognition

Data extraction from large quantities of information is a significant obstacle requiring innovative technologies. Natural language processing and other forms of information extraction necessitate specific pre-processing tools to evaluate the text's phonetic, syntactic, lexical, morphological, and semantic elements. Named Entity Recognition (NER) is an essential tool for pre-processing text, which has direct relevance in multiple natural languages applications such as Machine Translation, Information Retrieval, Automatic Text Summarization, and Question Answering, among others [88][89]. Named entities are words that can identify elements with similar properties to other elements. This is done using rigid designators, atomic elements, or members of the semantic class, which may vary based on the domain. This variety of domains such as categorizing content for news companies, improving customer service, and search algorithms. Its applications depend heavily on the domain, meaning that the process of identifying words also changes from case to case. For example, Biomedicine has entities like genes and their products; general domains have people, locations, organizations, numbers, times, and other variables, while homeopathic knows drug names and diseases as its own. These are all examples of important entities in the respective domains [90][88][91].

2.14 Deep Learning

Deep neural networks offer various applications, such as news classification, sentiment analysis, topic labeling, relation classification, question answering, and natural language inference. These models have demonstrated superior performance to conventional methods in speech recognition, image processing, and text understanding tasks [92]. Rapid advancements in deep learning enhance the performance of dialogue systems. Conversational systems and the tasks that fall within them extensively use deep learning architectures [93].

2.14.1 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is an advanced Recurrent Neural Network (RNN) form. This type of network incorporates forget gates which enable it to remember the state of the input for a more extended time than traditional RNNs; this enables it to process long sequences more accurately. Using these special forget gates, LSTMs hold advantages over traditional RNNs regarding memory and computation efficiency [94]. It consists of three gates – input, forget, and output – which work together to compute the hidden state according to specific equations [41]. An overview of LSTM architecture is shown in Figure 2.3. The equations are as below:

$$\mathrm{x} = \begin{bmatrix} \mathrm{h}_{\mathrm{t}-1} \\ x_{\mathrm{t}} \end{bmatrix} \tag{9}$$

$$f_t = \sigma(W_f.\mathrm{x} + b_f) \qquad \textit{Forget gate} \tag{10}$$

$$i_t = \sigma(W_i.\mathrm{x} + b_i) \qquad \textit{Input gate} \tag{11}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\,(W_c.X + b_c) \qquad \textit{Memory cell update} \tag{12}$$

$$o_t = \sigma(W_o.\mathrm{x} + b_o) \qquad \textit{Output gate} \tag{13}$$

$$h_t = o_t \odot \tanh\,(c_t) \qquad \textit{Output} \tag{14}$$

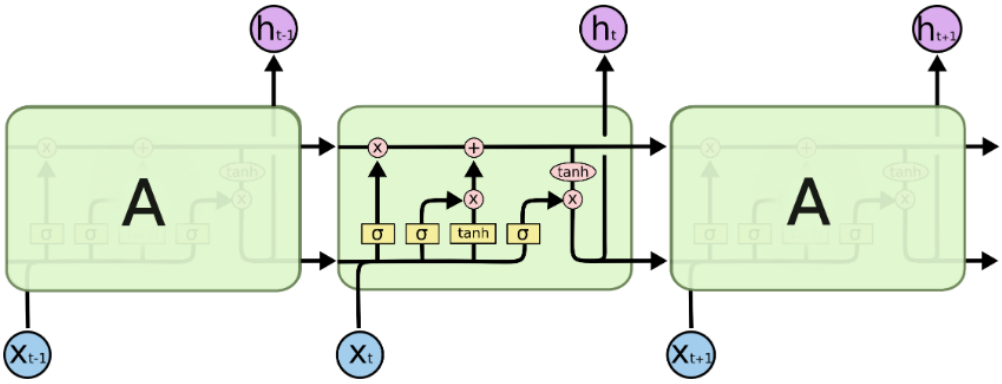

Figure 2.3: Overview of LSTM Architecture [95]

The LSTM process initializes with selecting which information to exclude from the cell state. A sigmoidal layer facilitates this decision called the "forget gate layer" based on factors such as h_{t-1} and x_t. This "forget gate" outputs a number in the range of 0 to 1 for each element in the cell state c_{t-1}. Decision-making regarding what information should be stored in the cell state comes next. This process is divided into two parts. Firstly, an input gate layer, comprising a sigmoid layer, determines which values are to be updated. Secondly, a tanh layer generates a vector of potential c_t values for inclusion in the state's data. Next, these two layers are put together to update the state. Thus update the old state into the new state c_t. The last step is output; a sigmoid layer is utilized to determine the divisions of the cell state that should be outputted. Afterwards, tanh shifts the values to a range of -1 and 1. The above procedure is completed by multiplying the applied tanh function with the output result of the sigmoid gate in order to emit only predetermined segments effectively[41] [94].

2.14.2 Bi-directional Long Short-Term Memory (BiLSTM)

The BiLSTM model is a type of neural network model that takes the strength of both the BiRNN and LSTM models. This model can propagate data in both forward and backward directions, which makes it more effective than single-directional models. Additionally, it can store current and past data for future use, making it even better suited for processing complex datasets. Figure 2.4 illustrates the information flow from both backward and forward layers in BiLSTM. This type of network is ideal for sequence-to-sequence tasks, such as speech recognition, forecasting models, and text classification [41].

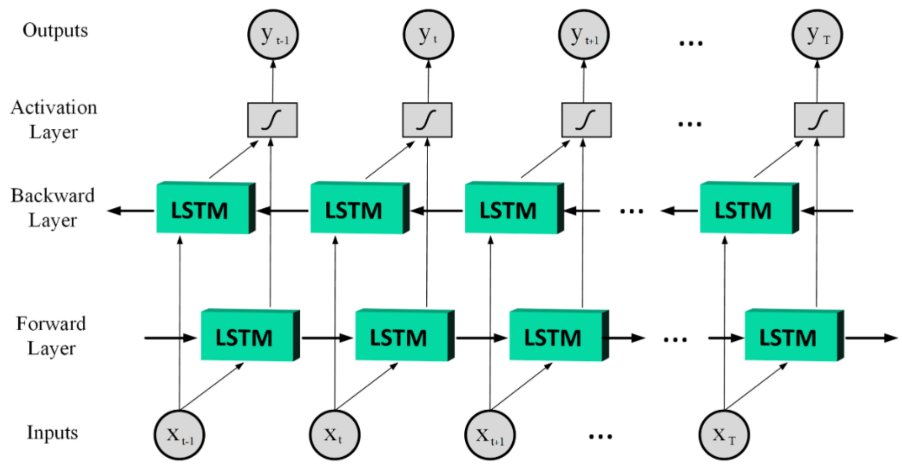

Figure 2.4: BiLSTM Architecture [95]

2.14.3 Sequence to Sequence (Seq2Seq)

Encoder-decoder models represent a significant advancement in deep learning. They effectively map sequences to sequences, making them well-suited for complex tasks such as time-series forecasting, generative chatbot, machine translation, and Image captioning. These networks improve traditional deep neural networks (DNNs), which can only be used on labeled data with vectors of fixed dimensionality. In addition, encoder-decoder models open up a whole new range of use cases by allowing the model to recognize and learn from structured or natural language inputs [96].

Seq2Seq or sequence-to-sequence modeling is a robust machine learning technique that takes a large set of sequence pairs as input and produces one by mapping the other. This model can be thought of in terms of input/output (I/O). For example, given an input sentence – "How are you doing?" – the output could be – "Not bad." [97]

The seq2seq model is an encoder-decoder model widely used as a base for other more complex solutions. In this architecture, an input sentence is first put through the "encoder," creating a context vector representing its meaning. Typically, Recurrent Neural Networks (RNNs) are specialized networks that take a sequence of inputs (x_1, ..., x_n) and generate an output by unfolding/unrolling the network. Each step consists of taking x and h_{i-1} to calculate a new hidden state, h. The vanilla implementation of recurrent networks is rarely used; however, due to the vanishing gradient problem - in long sequences, it can be challenging for RNNs to remember previous steps. To combat this issue, Long Short-Term Memory (LSTMs) have been implemented successfully[31][70]. Then, the encoder interprets and synthesizes the input sequence into internal state vectors and context vectors (in the case of an LSTM, these are called hidden state vectors or cell states).

In contrast, only the internal states are kept. The purpose of the context vector is to help integrate data from all inputs to support decoders so they can make accurate predictions [31]. Figure 2.5 present the architecture of the seq2seq LSTM model.

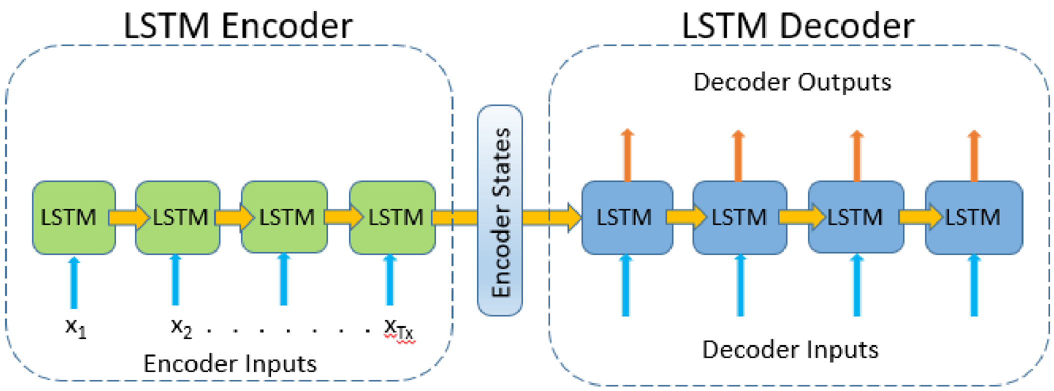

Figure 2.5: Basic Sequence to Sequence LSTM Architecture [98]

A Seq2seq model uses two distinct blocks, the encoder and the decoder, which are connected through a vector, referred to as the 'context vector.' This system facilitates direct communication between the two independent components for efficient data transmission.

Encoder: The encoder systematically considers every token in the input sequence and condenses its related information into a 'context vector' of fixed size. Once the entire sequence is analyzed, the encoder passes this encoded vector to the decoder for further processing.

Context vector: The context vector represents the whole meaning of an input sequence, allowing for more precise decoder block predictions.

Decoder: The decoder takes the context vector as input and then attempts to create the target sequence one token at a time. The decoder functions differently during the training and inference phases. The initial state of the decoder is set to be the same as the last state of the encoder.

Then, the decoder is primed with '<start>' to generate the next token, which will be the first word of the sentence. The teacher forcing technique is used to help guide the decoder toward predicting the correct output. This entails feeding it with the actual previous time stamp output instead of its predicted output for every time step until all words have been inputted and an '<end>' token has been predicted, signaling that a sequence has been completed. This '<end>' acts as a stopping condition during inference. In the inference phase, the decoder behaves differently than during training. A predicted output is used instead of taking the actual output as input from the previous time step. Aside from this difference in behavior, everything else remains the same [31] [70] [96] [97].

2.15 Evaluation Measures

In order to determine how effective a model is, evaluation measures are employed. There are numerous measures by which a model can be evaluated. Multi-class classification is a core task in machine learning involving more than two classes. Several performance indicators can aid in evaluating and comparing the different models or techniques used to undertake multi-class classification tasks. In addition, these metrics provide helpful feedback throughout the development process, such as when comparing the performance of two models or when tuning different parameters of the same model [99]. Many metrics used to evaluate the performance is described in the following subsections.

2.15.1 Performance Measure Analysis

Determining the quality of a classification model requires analysis of its accuracy and error rate. A classifier's performance can be evaluated using the Accuracy metric. It is determined by dividing the number of accurate predictions by the total number of predictions made. This provides an easy way to measure and compare a model's accuracy to other models [100]. The multi-class classification accuracy equation and error rate are shown below [101]:

$$Average\ Accuracy = \frac{\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{l} \tag{15}$$

$$Error\ Rate = \frac{\sum_{i=1}^{l} \frac{fp_i + fn_i}{tp_i + fn_i + fp_i + tn_i}}{l} \tag{16}$$

2.15.2 Confusion Matrix

The confusion matrix is an organized table that outlines the comparison of two raters in terms of true/actual classifications and estimated classifications. The row and column order for the classes should match so that correct classifications appear across the predominant diagonal from top left to bottom right. This corresponds to both raters agreeing on the result [99]. The matrix groups class results into four categories: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP reflects when both actual and predicted values are 1, TN denotes when both actual and predicted values are 0, and FP applies when the actual value is 0, but the predicted value is 1. FN represents a situation when the actual value is 1, but the predicted value is 0 [102]

[103]. Figure 2.6 shows the confusion matrix for multi-class classification.

		PREDICTED classification			
	Classes	a	b	c	d
ACTUAL classification	a	TN	FP	TN	TN
	b	FN	TP	FN	FN
	c	TN	FP	TN	TN
	d	TN	FP	TN	TN

Figure 2.6: Confusion Matrix for Multi-class [99]

- **Precision and Recall and F1-score and Specificity**

  The Precision, Recall, and F1 scores are important metrics often used when dealing with imbalanced datasets (i.e., datasets with most samples belonging to one class). Specifically, these metrics are used instead of accuracy or error rate to measure the model's performance on such datasets. For example, precision is the ratio between true and predicted positives and measures how precise or accurate a model makes predictions. Specificity measures the probability that an individual tested negative will be appropriately identified as negative [103].

$$Precision_M = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l} \tag{17}$$

$$Recall_M = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l} \tag{18}$$

$$Fscore_M = \frac{(\beta^2 + 1) Precision_M Recall_M}{\beta^2 Precision_M + Recall_M} \tag{19}$$

Where β is a parameter that used to tune the relative importance of precision and recall. β=1 gives equal weights to precision and recall.

2.15.3 Cross Entropy

Cross-entropy is a powerful tool for measuring the similarity between two probability distributions. It is often used as a loss function in classification tasks, effectively comparing predicted classes and ground truth values [99]. For multi-class classification, categorical cross entropy is used. The neural network outputs are first passed through a softmax activation function. This produces a vector of probabilities for each of the input classes. The highest probability will represent the predicted class for the data [104]. Multi-class cross entropy loss is defined in equation 20.

$$CE(t,p) = -\sum_{i=1}^{S}\sum_{j=1}^{N} t_{ij} \log (pij) \tag{20}$$

2.15.4 K-fold cross Validation

Evaluating an Artificial Neural Network (ANN) model's quality requires choosing the finest set of parameters, including number of hidden nodes and back-propagation learning rate. Unfairly dividing training and testing datasets so none become representative of all data can lead to a below-par estimate of a Deep Learning model's reliability. The k-fold cross-validation procedure helps here, as it obtains evaluation results for multiple neural nets by computing their average score; then reports its performance measure for the whole run. This process includes repeating training and testing K times over various samples from the data set – which is somewhat computationally expensive, though valuable when data samples are few in number. In this cross-validation technique, each time K distinct models are implemented: one part of the dataset is held for assessment purposes while the remainder is utilized for training. Finally, the error rate is computed by aggregating overall results [105][106]. Figure 2.7 Present cross validation with 5-fold.

Figure 2.7: Cross Validation with 5-Fold [107]

2.15.5 BiLingual Evaluation Understudy (BLEU) Score

BLEU (BiLingual Evaluation Understudy) is an evaluation model developed by Kishore Papineni et al. in the 2002 paper "BLEU: A Method for Automatic Evaluation of Machine Translation." This algorithm was designed to compute the quality of output text generated by a machine translation, although it has since been adopted for use in chatbot evaluation. BLEU tokenizes a conversation candidate and compares it to an existing reference using n-grams from 1-4.

The BLEU score value within the interval [0, 1] defined in Equation 21 [70]:

$$BLEU = BP \times \left(\prod_{n=1}^{N} precision_n\right)^{1/N} \quad (21)$$

where N is a maximum n-gram number, n = [1, N]. BP (brevity penalty) and $precision_n$ are defined with Equations 22, 23 respectively.

$$BP = min\left(1, exp\left(1 - \frac{ref_{length}}{out_{length}}\right)\right) \tag{22}$$

$$precision_n = \frac{\sum_n min(m_{out}^n, m_{ref}^n)}{\sum_{\acute{n}} m_{out}^{\acute{n}}} \tag{23}$$

Where:

- ref_{length} is a reference length,
- out_{length} is the chatbot output length,
- m_{out}^n is the number of n-grams in the chatbot output matching the reference,
- m_{ref}^n is the number of n-grams in the reference,
- $\sum_{\acute{n}} m_{out}^{\acute{n}}$ is the total number of n-grams in the chatbot output.

Depending on the score, its interpretation can range from useless (BLEU < 10) to better than human (>60). Ultimately, this approach measures how accurate and well-formed chatbot responses are[108] [70]. These values and interpretations were taken from Google Cloud's AI and Machine Learning product description in Reference [109].

CHAPTER THREE

PROPOSED METHODOLOGY

3.1 Overview

This chapter presents the details of the proposed work which involve two models, the first is to improve smart marketing chatbot (SMC) responses and accuracy using deep learning. The second model is to understand the user request, including intent classification using machine learning and deep learning, and extraction entities using deep learning to understand what the user asks about, which is planned to combine this two models in future work.

Before covering the two proposed models, referring to the necessary steps of dataset preprocessing is essential. Then the models will be explained in detail.

3.2 Preprocessing The Amazon Office products Dataset

To increase the accuracy of the models, some preprocessing is required to implement. Below is several preprocessing that have been implemented.

1. *Reduce sequence length*: Reduce the sequence length of answers, which often exceeds 100 words. This procedure is performed to handle data outliers, which consist of a large number of sentences with an abnormally large number of words, and to prevent training process errors.
2. *Lowercasing:* This step mean lowercasing all the pairs of questions and answers.

3. *Word formalization:* Each word abbreviated or merged is transformed into its regular form.
4. *Tokenization:* Breaking up the given sentences into separated words called tokens.
5. *Cleaning data:* Some special characters in the dataset are omitted because the model focuses on word generation. In addition, duplicates, mentions, and links were also removed.
6. *Remove stop words:* Removing some stop words, a custom list of stop words to be removed is created. Figure 3.1 shows the custom list of stop words.

```
stopwords =['the','this','to','a','it','i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves',
            'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves',
            'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its',
            'itself', 'they', 'them', 'their', 'theirs', 'themselves','or', 'as', 'until', 'while', 'of',
            'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before',
            'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under',
            'again', 'further', 'then', 'once', 'here', 'there', 'all', 'any', 'both', 'each', 'few', 'other',
            'some', 'such', 'so', 'than', 'too', 'very', 'those', 'on', 'yourselves','alone','he', 'it', 'see',
            'than', 'already', 'our','that', 'its', 'for', 'of', 'others', 'the', 'hence', 'something', 'hereby',
            'his','their', 'by', 'namely', 'she', 'become', 'a', 'using', 'him', 'or', 'me', 'we', 'you', 'them',
            'your', 'and', 'an', 'this','these','there','be','if',"i'm"]
```

Figure 3.1: Custom Stopwords List

7. *Lemmatization:* This step is implemented only on answers; the reason behind not implementing the lemma on questions is that it transfers some questions' words to their actual root.
8. *Normalization:* Performed on all data *such as don't* be *do not*, *I'll* be *I will*, etc.
9. *Spell correction:* For correction of the spells, the spelling correctness is performed.

10. *Remove unrelated data:* Unrelated responses *such as* “I have same question”, “emoji’s ”, “ I don’t know” , etc are removed from the dataset.
11. *Cosine similarity:* reduce question similarities and retain unique questions/answers with a 0.8 score, this score is suitable for datasets, especially for short sentences. Figure 3.2 shows the example of cosine similarity.

123	0.8660254037844388	is made plastic metal	222	is metal plastic
123	0.8660254037844388	is made plastic metal	7146	is made metal
123	0.8944271909999159	is made plastic metal	9746	is arm made plastic metal
123	0.8660254037844388	is made plastic metal	10474	is metal plastic

Figure 3.2: Example of Cosine Similarity

These steps are essential to improve the model's performance. After preparing the dataset, it is ready to be fed into models.

3.3 Smart Marketing Chatbot Improvement

This section will cover the methodology of building and improving a generative-based chatbot. For this purpose, the Keras Functional API is used to construct an LSTM seq2seq model as a basis for building an office Chatbot. The Chatbot should be able to answer questions posed by users effectively and with satisfactory precision. The subsequent steps for implementing the model are described as follows. Figure 3.3 shows the block diagram of SMC model.

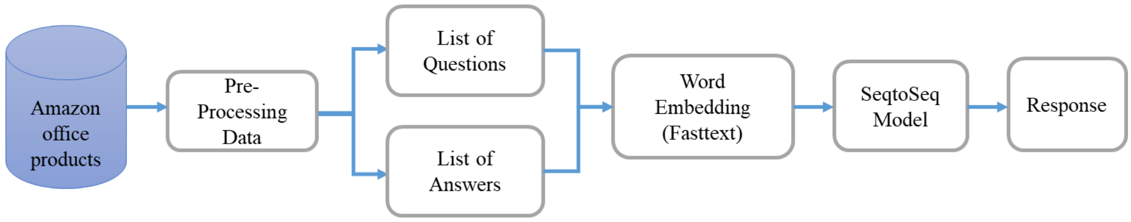

Figure 3.3: Block Diagram of SMC Model

The architecture of sequence to sequence model is presented in Figure 3.4.

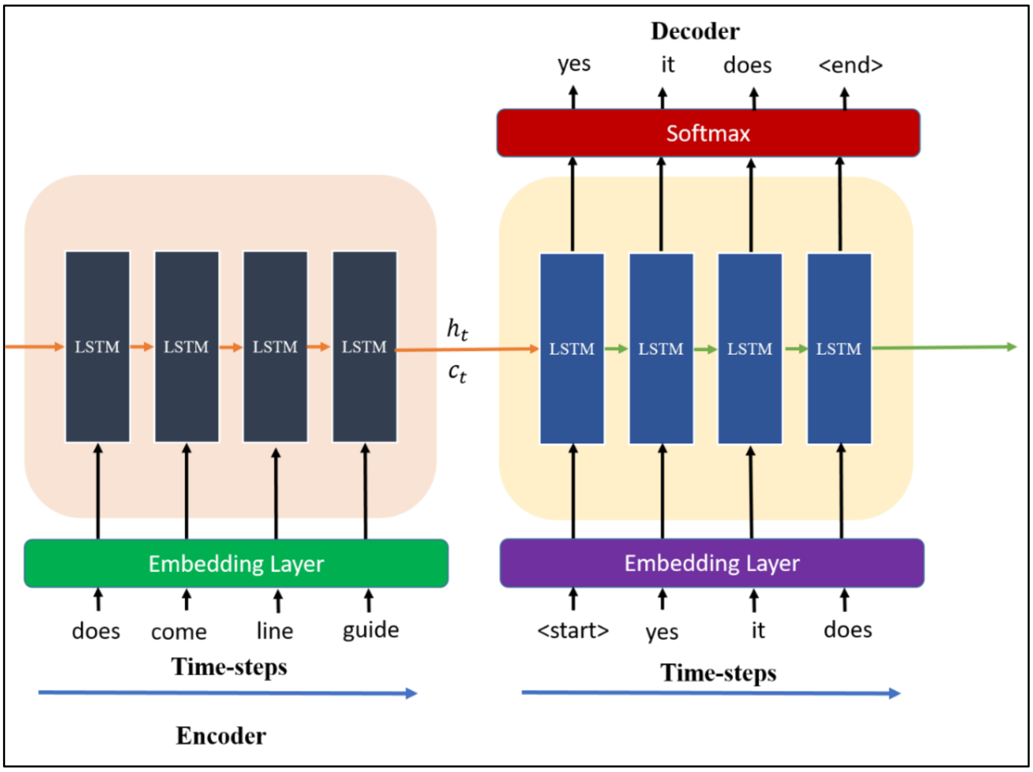

Figure 3.4: Seq2Seq Model Architecture

3.3.1 Data Preparation for Seq2Seq Model

First, the preprocessing steps as mentioned in section 3.2 are implemented on the data, also rare words from answers are removed. After that, the conversations are divided into two different lists - questions and responses which questions are converted to a list also the matched answers to questions are converted to another list. Tokenizer is created and both lists of sequences (questions - answers) are tokenized then the whole vocabulary of both lists is loaded into the vocabulary list. Next step, the special symbols <start> and <end> append to the begin and end of all the target sequences, which these symbols help in the training and inference phase. In addition, both sequences are padded to their max length. After creating the vocabulary, each word in the sequence transformed into numbers. The input sequence must be one-hot encoded in order to be processed properly. In addition, the output sequence of the decoder should also be one-hot encoded and the "start" token added before each word, which will signal the decoder to begin decoding. Thus, the model requires turning into three arrays named *encoder_input_data*, *decoder_input_data*, and *decoder_target_data*. The *encoder_input_data* contains a one-hot vectorization of questions. *Decoder_input_data* contains a one-hot vectorization of answers. *Decoder_target_data* is the same as *Decoder_input_data* but with eliminating the first token <start> from it.

The pre-trained fasttext is used in this model for word embedding. Due to the one-hot encoding's tendency to produce rather large dimensions, the embedding layer reduces the size of the input word vectors. A more effective technique to express words is with an embedded vector representation. The word-level-based model produces the output words at a time after processing the input words at a time.

3.3.2 Chatbot with Sequence to Sequence Model

The neural network model has two input layers, two embedding layers, and two LSTM layers for Encoder and Decoder.

The input layer is a two-dimension that passes the sample number and the max sequence length of data to the embedding layer. Next embedding layer takes two-dimension (max sequence length, vocab size) as input. It passes three-dimension (number of samples, max sequence length, embedding dimension) to the encoder layer as output.

The encoder part takes one word of the sequence at each time step, processes it, and encodes that information into a final hidden state h_t and cell state c_t. These internal states are then passed on to the decoder part of the model, which uses them to generate the target sequence. The dropout is used to avoid overfitting. The critical argument of the LSTM layer at the encoder phase is the return state which returns the internal state which contains the cell and hidden state, and the output state, where the last one is discarded.

After processing the entire input sequence, the encoder passes its internal states to the decoder section, where predicted output generation begins. LSTM decoder block takes the initial states (h_0 and c_0) from the encoder's final states (h_t and c_t), which act as the ‘context’ vector for aiding the decoder in producing the desired target-sequence. The decoder model is trained using Teacher Forcing; in teacher forcing, instead of providing the predicted output from the previous time-step, it is supplied with the true output. This means that the input for the next time step is not fed by prediction output but with the proper output. The important argument of the LSTM layer at the decoder phase is return sequence that returns all hidden states of all time-steps.

At this point, the error for each predicted output over time is calculated and backpropagation is used to update the model's parameters. The categorical cross-entropy loss function is applied between the true target and predicted sequences to calculate the final loss.

After this process, the decoder's final states are discarded from the model. The model is trained with several epochs with Adam optimizer and categorical_crossentropy loss function. The following Table 3.1 presents the model summary. Figure 3.5 Illustrate model visualization.

Table 3.1: Present Seq2Seq Model Summary

#	Layer (type)	Output Shape	Params	Connected to
1	Input_1 (Input Layer)	[(None,21)]	0	
2	Input_2 (Input Layer)	[(None,21)]	0	
3	Embedding_1 (Embedding)	(None,21,300)	2940300	Input_1[0][0]
4	Embedding_2 (Embedding)	(None,21,300)	2940300	Input_2[0][0]
5	Lstm (LSTM)	[(None,300),(None,300),..]	721200	Embedding_1[0][0]
6	Lstm_1 (LSTM)	[(None,21,300),(None,300),..]	721200	Embedding_2[0][0] Lstm [0][1] Lstm [0][2]
7	Dense (Dense)	(None,21,9801)	2950101	Lstm_1[0][0]
Total params:		10,273,101		
Trainable params:		4,392,501		
Non-trainable params:		5,880,600		

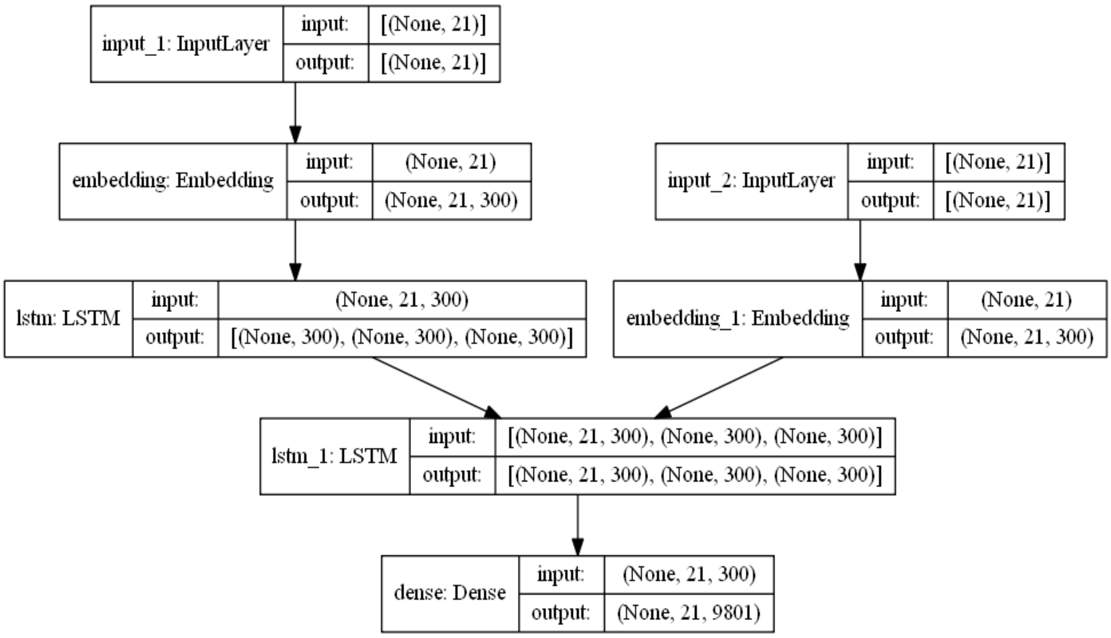

Figure 3.5: Seq2Seq Model Visualization

3.3.3 Chatbot Inference Model

Once training is finished, inference models with encoder and decoder components are created. Inference take place to assess the accuracy of the output from the model. The encoder block is the same for testing in both training and testing phase, but the decoder block has some differences. The predicted output from this time-step rather than a true output (unlike in the training phase) is used as input for the subsequent time-step. Besides that, everything else remains unchanged compared to the training phase.

The encoder model takes the question as an input and outputs long-short-term memory (LSTM) states (h & c). The decoder model takes in two inputs: the LSTM states (output of the encoder model) and a sequence of answer inputs that do not include the <start> tag. This output answers the question fed into the encoder model and its state values.

The process starts by converting string questions to integer tokens with padding. The encoder model uses the input question to predict the state values. These state values are then used as parameters for the LSTM of the decoder model. Subsequently, a sequence containing the <start> element is generated as input and passed into the decoder model. Then the <start> element is replaced with the element predicted by the decoder model and updated its state values accordingly. The same steps will be repeated until the sequence hits the <end> tag or reaches its maximum answer length.

3.3.4 BLEU Score

BLEU score is used to evaluate the quality of chatbot responses to user's input. BLEU scores are calculated based on n-grams between the candidate response of the chatbot reply and the reference to an existing, accurate reply. The BLEU scores are calculated using the python implemented *bleu_score* module from the *translated* package in the *nltk* platform. To generate the BLEU scores for each experiment of the n-gram, the value of weights in the BLEU algorithm needs to be set. The weights used to calculate the BLEU scores for each n-gram experiment are given in Figure 3.6.

1-gram BLEU weights = (1, 0, 0, 0)

2-gram BLEU weights = (0.5, 0.5, 0, 0)

3-gram BLEU weights = (0.33, 0.33, 0.33, 0)

4-gram BLEU weights = (0.25, 0.25, 0.25, 0.25)

Figure 3.6: BLEU N-gram Weights

3.4 User's Request Understanding Model

To understand the user's request, two models were proposed, one to know the user's intent and the other to extract the entities and know what the user asks. Figure 3.7 shows the understanding of the user's request models.

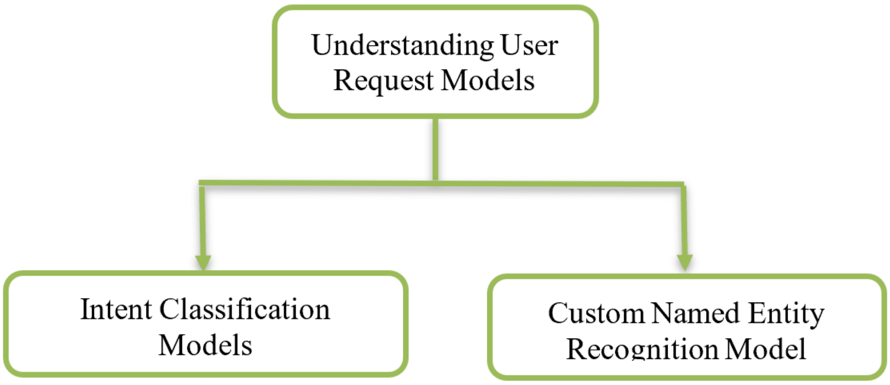

Figure 3.7: Understanding User's Request Models

3.4.1 Intent Classification Models

Intent classification is the first model in understanding an effective chatbot. This problem is modeled as a multi-classification with a single label and output vector of class probabilities. By recognizing and studying these probabilities, the bot's response can fine-tune to offer an appropriate answer every time. Figure 3.8 shows the block diagram of the proposed intent classification system.

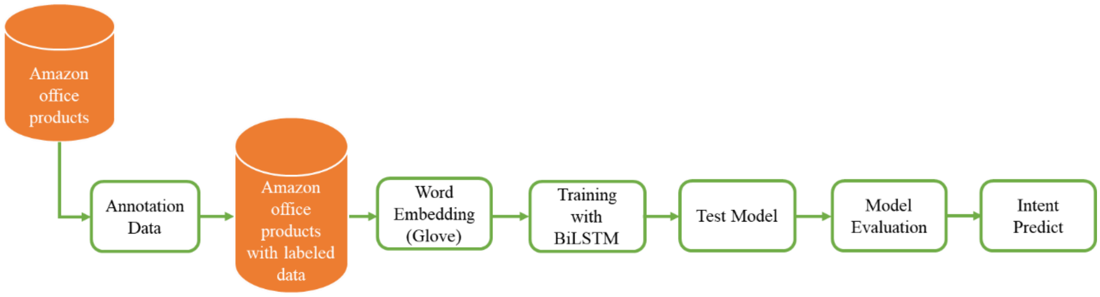

Figure 3.8: Block Diagram of Proposed Intent Classification System

3.4.2 Annotation Data

To begin designing and implementing the intent classification models, the dataset should prepared because the proposed dataset doesn't contain an intent label for each utterance. The labels of intents are identified according to questions words. Identifying intents consisted of attempting various techniques, such as keyword matching and topic modeling. However, these methods were ineffective since the data is similar and close together, which causes overlap between the classes. Therefore, using the question words to label the dataset is the appropriate way to take it. To label the data, a regular expression is applied to the questions' text, and then nine classes are extracted and given the following labels: *how*, *what*, *auxiliary-can-could*, *auxiliary-is-are*, *auxiliary-do-does-did*, *auxiliary-have-has*, *auxiliary-will-would-should*, *5-wh* (*when*, *where*, *which*, *why*, *who*), and *unknown*. Hence, the unknown class is removed from intents because it affects the model's performance and causes ambiguity in the intent classifier, so the dataset is classified with eight classes. Table 3.2 shows the number of samples in each intent class.

Table 3.2: Number of Samples in Each Intent Class

N	Intent	Number of Samples
1	auxiliary-do-does-did	3036
2	auxiliary-is-are	2447
3	auxiliary-can-could	1938
4	how	1519
5	auxiliary-will-would-should	1421
6	what	1307
7	5-wh	460
8	auxiliary-have-has	104

3.4.3 Intent Classifier using Machine Learning Techniques

In this thesis section, machine learning techniques are explored to address the issue of classifying intents for dialog systems. The performance of four classification methods based on machine learning is tested and the performance of the methods are compared. These approaches are Support Vector Machine (SVM), Random Forest, Naive Base, and Logistic Regression.

Data augmentation generates new equivalent synonym utterances for each question to achieve more utterances for minority classes in the dataset. Accordingly, contextual embedding BERT from the *nlpaug* library fed the surrounding words to find the most appropriate synonym word for augmentation. The augmented data implemented on the two minor classes containing to increase their utterances by generating new synonym text. Samples of augmented data are shown in Figure 3.9.

Sample of Augmenting Text

Original Text:

does the canon mf 8380cdw have a legal-sized scan glass?

Augmented Text:

1) but does the Russian canon mf 8380cdw have also a low legal sized standard scan glass?
2) also does the canon mf canon 8380cdw have a top legal-sized micro scan rate glass?
3) does the 1996 canon standard mf 8380cdw have a single legal body-sized alpha scan glass?
4) what do the canon models mf and 8380cdw have a legal recognition-sized scan for glass?
5) does however the canon mf canon 8380cdw lens have on a legal-sized scan d glass?

Figure 3.9: Samples of Augmented Data

Figure 3.10 shows the intent classification model using machine learning techniques.

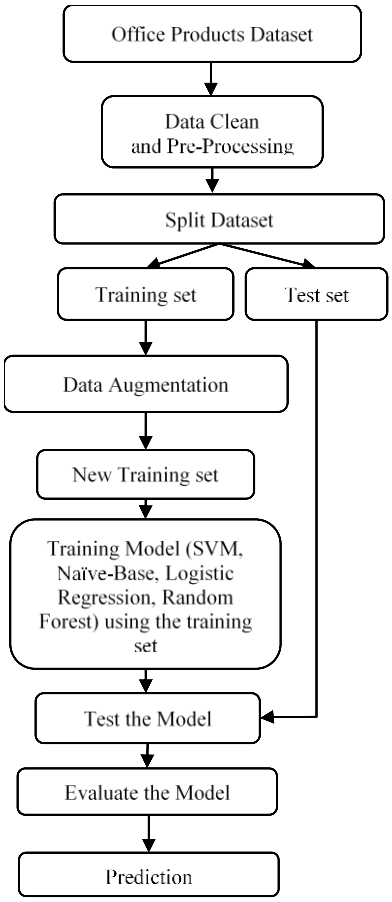

Figure 3.10: Intent Classification Model using Machine Learning Techniques

3.4.4 Intent Classifier Using Deep Learning

In this thesis, the BiLSTM model is used for intent classification which improved the results of machine learning classification techniques.

Bi-Directional Long Short-Term Memory Units (BiLSTM) have the advantage of processing input in both directions, past and future information. Compared to uni-directional LSTMs, BiLSTM can provide more context, increasing accuracy and overall problem-solving effectiveness.

It is important to take specific actions to get the input text ready for training. The dataset is split into the train and test sets by 80% and 20%, respectively. Next, the label encoder is used to label target data. Questions are tokenized and converted into a sequence of integers using the tokenizer. The max length of sentences is equal to 121 characters. Hence, the sequences are padded to be the same length for modeling as Keras requires. Standardizing the inputs' sizes will ensure that different dimensions don't mess up the model. Next, word embedding is extracted to generate an embedding matrix. For this model, Glove is used. The word embedding extracted from Glove is contained in embedded index. A for loop searches the embedded index for words in office products vocabulary (word index) and attaches those vector values to the embedding matrix. A co-occurrence matrix contains word values and their relationships to other words for the dataset. Here, the word index lists all the unique words (vocabulary) in the training set. The glove size is glove.6B.200d. Therefore, the embedding dimension is the same size as the glove dimension and is equal to 200D. Table 3.3 shows details of the dataset.

Table 3.3: Dataset Details

Feature	**Count**
Training set	9976
Testing set	2495
Vocabulary size	7329
Num. of Intent Classes	8
Embedding dim	200
Max_length	121

Now the data is ready to be fed into the BiLSTM model. A sequential model is used for this part of the work. The first layer is the Embedding layer, which uses a vector of fixed length 121 and an input dimension equal to the vocabulary size to represent each word. The embedding layer is seeded with glove word embedding weights. Therefore, the learned word weights did not update in this model and the trainable attribute for the model is set to be False. The following layer BiLSTM layer with its hyperparameters is added. Next, the dense layer with the relu activation function added. Following, also another dense layer, like the previous one is added.

Additionally, a dropout layer is added. The last layer is the output layer contained the number of intent classes and softmax activation function. The model compiled with the Adam optimizer, categorical cross-entropy loss, and performance metrics accuracy. Figure 3.11 shows the summary of the proposed BiLSTM intent classifier model. Figure 3.12 shows the architecture model of intent classification.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 121, 200)          1417000
_________________________________________________________________
bidirectional_1 (Bidirection (None, 256)               336896
_________________________________________________________________
dense_3 (Dense)              (None, 600)               154200
_________________________________________________________________
dense_4 (Dense)              (None, 600)               360600
_________________________________________________________________
dropout_1 (Dropout)          (None, 600)               0
_________________________________________________________________
dense_5 (Dense)              (None, 8)                 4808
=================================================================
Total params: 2,273,504
Trainable params: 856,504
Non-trainable params: 1,417,000
_________________________________________________________________
```

Figure 3.11: Summary of Proposed BiLSTM Intent Classifier Model

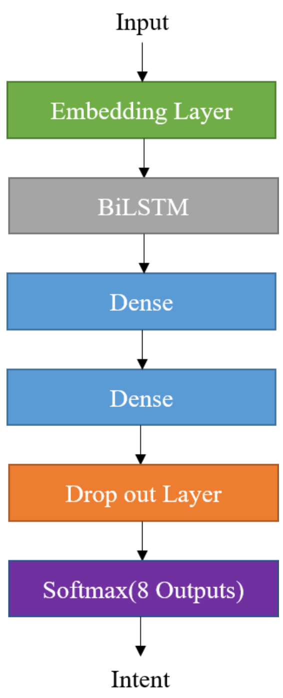

Figure 3.12: BiLSTM Intent Classifier Architecture Model

The model fitted on the training data with a batch size 64 for 50 epochs. Additionally, 20% of the training data was used for validation, as specified in val_split. At the end of each epoch, performance metrics were evaluated using this validation data.

3.4.5 Custom Named Entity Recognition Model

While parsing a user utterance, it is necessary to determine the user's intent and extract the parameters from the required text to fulfil the request. These factors in the field of NLP referred to entities. Named Entity Recognition (NER) is a rapidly expanding subfield of NLP that tries to find and label relevant information in texts with high precision.

Entity recognition is a multiclass token classification that assigns a class to the input sequence. By default, standard NER looks for predefined entities in a text such as location, person's name, language, organization, event, etc. This standard model trained on predefined labeled data and then used to extract information from an unlabeled dataset. Standard NER isn't applicable in this thesis because it can't extract relevant information based on the specified domain. Thus, a custom-named entity created to extract relevant entities from office products such as *action*, *color*, *brand*, *material, etc*. The following subsections show how custom NER putted into place. The block diagram of the proposed custom entity recognition shown in Figure 3.13.

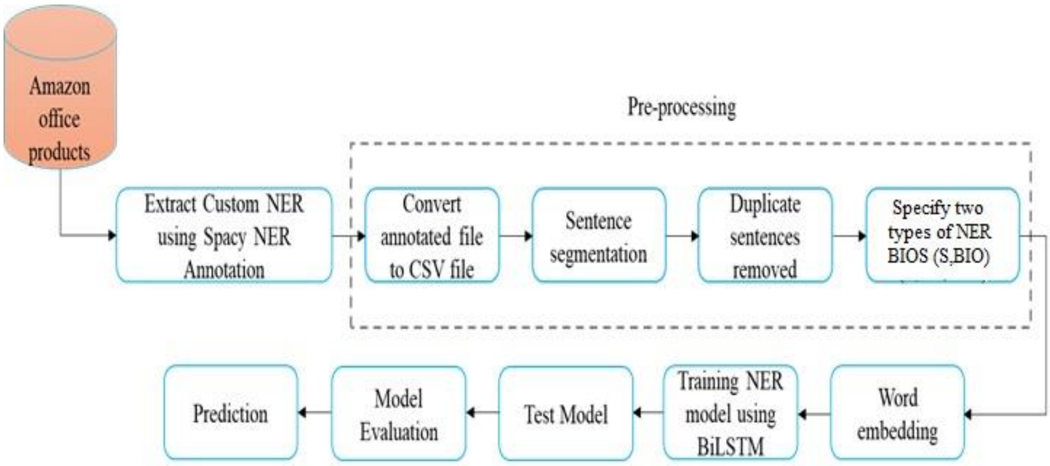

Figure 3.13: Block Diagram of Custom Entity Recognition System

3.4.5.1 Extract Custom NER Using Spacy NER Annotator

To create custom NER for office products should extract and specify each entity and its tag, for this purpose to produce the SpaCy supported annotated data format, a spacy NER annotation tool by Agateteam is used. The annotation of a sentence sample is shown in Figure 3.14.

Figure 3.14: Annotation of a Sentence Sample

After annotation data is completed, the output file is in *json* format and the entities are derived from their locations and extracted into a CSV file. Figure 3.15 presents the *json* file of spacy annotator. The number of extracted labels are 14 and labels are shown in Table 3.4.


```
("does phone have caller id ",{"entities":[(5,10,"Object"),(11,15,"Action"),(16,25,"Title")]}),
("does phone have visual indicator when ringing ",{"entities":[(5,10,"Object")]}),
("can access messages remote location like vacation am away phone ",{"entities":[]}),
("does phone have headphone jack ",{"entities":[(16,25,"Title"),(5,10,"Object")]}),
```


Figure 3.15: Json File of Spacy Annotator

Table 3.4: Label List

N	**Named Entity**	**Count**
1	Object	9540
2	Action	5525
3	Title	5435
4	Number	4382
5	Color	814
6	Materials	782
7	Software	670
8	Time	513
9	Size	297
10	Cards	192
11	Organization	184
12	Brand	173
13	Nationality	154
14	GPE	129

3.4.5.2 Preprocessing Custom NER Model

The dataset of custom NER comprises questions and corresponding entities to each question. In addition, each question repeated based on the number of entities that possess it. In order to implement the model, it is necessary to segment each sentence into segmented words and each word indexed with its corresponding sentence. Thus, each segmented token corresponds to its tag. Therefore, duplicated sentences was removed and each sentence appears with its entities.

In this thesis, BIOS notation is used for sequence labeling of NER tasks, where B stands for the beginning of entity, I inside, O stands for outside, and S stands for single. The single-tagged used for the entity that is consisting one word. The O-tagged token is not classified as an entity type, and the model assigned no label to it. For example, in the sentence "can print a very label," entity tags will be as follows "O S-Action O O S-Object". The overall number of sentences that contain entity is 12197; the sentences formed as tuples consisting of tokens and its tags. The next step is constructing a dictionary (word2index) that assigns a unique integer value to each word in the corpus.

Moreover, reversed dictionary (index2word) is constructed. Note that one extra entire padding is included for the end of sentence. Now the dataset has been indexed and needs to pad all sentences to the maximum length of occurred sentence in the corpus.

The last transformation is one-hot encoding the labels. At last, the dataset is split into train set and test set by 80% and 20% respectively. Table 3.5 shows some details of preparing the custom NER model.

Table 3.5: Details of Custom NER Model

Features	**Count**
Number of sentences	12197
Unique tags	20
Unique Entities	1741
Max length	25
Num_words	9232
Training set	9757
Test set	2440

3.4.5.3 Custom NER with BiLSTM Model

In this thesis, BiLSTM is used for implementing a custom-named entity. The BiLSTM model generates vector representations of words within a sentence. This model processes each word in the forward and backward direction, allowing it to access contextual information from both preceding and following words to create its vector representation. Different layers are used to build the model the first one is the input layer. The shape parameter of the input layer is a tuple that specifies the number of dimensions present in the input data. The next layer is the embedding layer which consists of three parameters the input dim, output dim, and input length; each parameter presents the vocabulary size, the embedding dimension, and the max sequence length respectively. The next layer is the BiLSTM with two parameter units that indicate the dimensionality of output space, and return sequence equal to true that returns the whole output sequence. In addition, the Time Distributed Layer is used to apply a dense output layer to processing each element within a sequence. The Adam optimizer is used. The model fitted on the training data with a batch size 64 for 200 epochs.

Additionally, 20% of the training data was used for validation, as specified in val_split. At the end of each epoch, performance metrics were evaluated using this validation data. The model architecture of implementing BiLSTM on custom NER is shown in Figure 3.16.

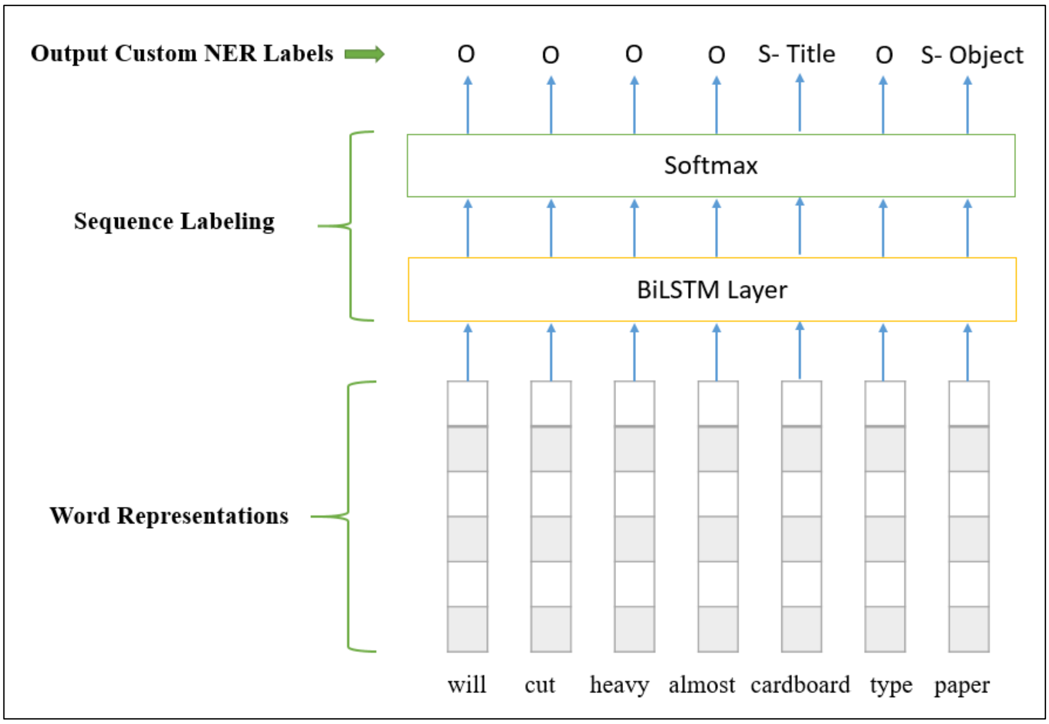

Figure 3.16: Architecture of Custom NER with BiLSTM

3.4.6 Evaluation Measures

In this phase, the presented system is evaluated utilizing a testing dataset to ascertain accuracy. To ensure the precision of results, cross validation was utilized. The metrics employed included cross entropy, accuracy, recall, F1-measure, specificity, and precision. The training and testing process was conducted ten times with ten folds used in each iteration. The cross-validation procedure is a powerful tool for assessing model performance on unseen data. It involves the evaluation of a machine learning model on a limited sample dataset known as resampling under one parameter, referred to as 'K'. This allows us to estimate how the model will perform on unseen datasets during training. The number of K-fold used is 10 fold. Each K's dataset is split into train, validation, and test. The result represented by computing the average of the performance.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Overview

The thesis objectives are to improve a smart marketing chatbot (SMC), understand user requests, including intent classification using machine learning and deep learning, and extract the entities using deep learning to understand what users ask. These objectives demonstrated in the previous chapter. This chapter aims to test the end-to-end effectiveness of the proposed systems with varying parameters. All systems are evaluated independently. The following sections discuss system requirements and information details regarding the dataset. Meanwhile, results from each model starting with SMC, intent classification models with machine learning and deep learning methods, and named entity recognition are listed and discussed in the following sections respectively.

4.2 System requirements

For machine learning and deep learning to be performed on any dataset, the software/program must utilize a computer system with sufficient computational capacity. Hence, the proposed system is implemented as follows:

Processor: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz

RAM: 16.0 GB

Hard Disk: 1TB HDD + 512 GB SSD

Graphics Processing Unit (GPU): MX450

Operating System: Windows 10 – 64bit

Programming language: Python

4.3 Dataset

The Amazon office products are gathered from various product pages on Amazon to create the dataset employed in this thesis. There is some processing going on, as well as the addition of features like labels to questions. The total dataset is divided into train 80% and test 20%. In addition, the training dataset is divided into 80% train and 20% validation sets. Figure 4.1 present a single response of the used dataset before performing the preprocessing steps. As shown in the Figure 4.1 the sequence length exceed 1000 words. While Figure 4.2 presents dataset samples after processing and adding intent labels to them. Table 4.1 shows the dataset division rates after processing.

Figure 4.1: One Response Sample Before Preprocessing The Dataset

	ASIN	Intent	question	Answers
0	439537886	what	what do use write leaves can wiped off	dry erase or wet erase markers
1	439537886	auxiliary_can_could	can anyone tell measurement leaves much	they are around the size of the palm of your hand
2	439537886	how	how do leaves attach tree	add a little tape on the back
3	792293347	_6_wh	which one options are laminated its not clear ...	The traditional one is laminated
4	1609964152	auxiliary_is_are_it	am missing days week number one calendar is an...	make them or look on Teacherspayteachers.com
5	1609964152	how	is really only 8 inches across individual numb...	Its 25 and a half width x 35 inches length
6	1609964152	auxiliary_is_are_it	is pockets numbers standard size number cards ...	Yes, I've switched the cards throughout the year
7	1609964152	auxiliary_do_does_did	does have cards weather	no
8	B00001U0S6	auxiliary_is_are_it	is sound quality good	no
9	B00001U0S6	auxiliary_is_are_it	is cordless phone	no
10	B000034DLQ	auxiliary_do_does_did	does laminate hot cold pouches	I have used it with the cold and it has worke...
11	B00004TS2I	auxiliary_is_are_it	is paper thick enough withstand coloring color...	yes but probably not Magic markers

Figure 4.2: Dataset Samples after Processing

Table 4.1: Dataset Division Rates

Dataset Name	Intent Classes	Entity Tags	Train Dataset	Test Dataset	Val Dataset
Amazon Office Products	8	20	9826	2457	2547

4.4 Smart Marketing Chatbot Evaluation

This thesis aims to improve a smart marketing chatbot that can provide the required support for customers and help them find answers to FAQ and other questions regarding office products interactively. In addition, provide generative-based answers not only depending on predefined answers.

Several experiments and stages are implemented to obtain satisfactory results to build a robust and accurate chatbot. The following experiments show the improvement procedure for designing the intelligent chatbot.

Experiment 1:

The first experiment is implemented on the dataset with pair of questions and multi-answers with a sequence length of 150 tokens. Indeed, the responses length was too long over 300 tokens. The hyperparameters used to implement this experiment are shown in Table 4.2. model accuracy and loss curves are shown in Figure 4.3.

Table 4.2: Hyperparameters of Experiment 1

LSTM unit	dropout	Recurrent dropout	Output dim	Learning-rate	Batch size	Epochs
256	0.5	0.5	256	Adam(0.0001)	64	300

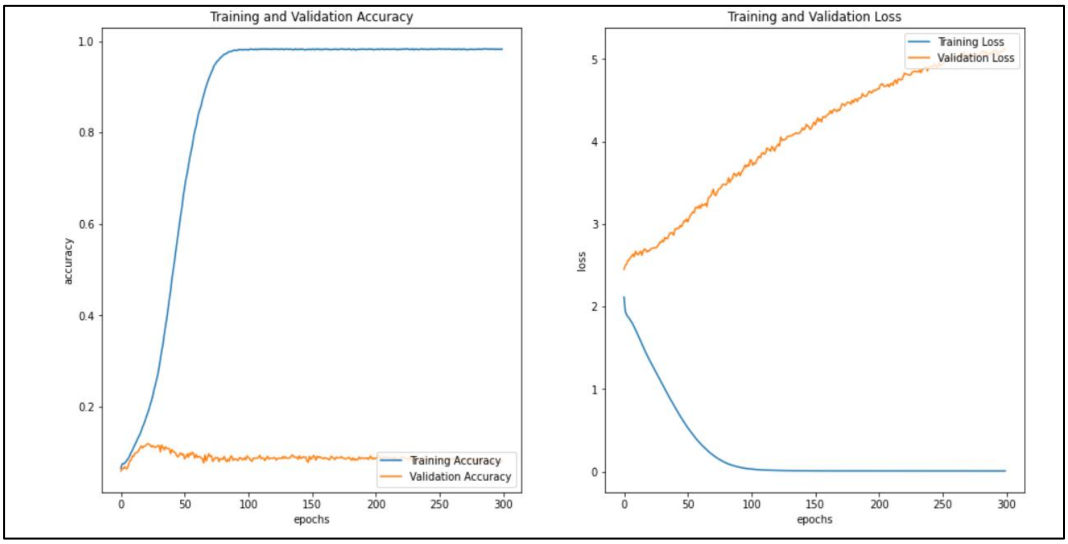

Figure 4.3: Accuracy and Loss of Experiment 1

The vocabulary size and the sequence length directly affect the model's performance. The results of this experiment model are not satisfied because large vocabulary size and long sequence length. A conversation between the chatbot and the user is presented in Figure 4.4.

```
Enter question : hello
 re designed found at t roll both lines honestly do not believe it is honestly for tabs honestly do they do honestly do not do
honestly when you cant answer as well as thick as a standard answer honestly i dont know honestly i cant help honestly i cant s
ay it is honestly for as i have
Enter question : hi
 to this pen i had large 1 year and a half inch i do not tell that it wouldnt honestly am a very long i am sorry but i dont kno
w its not a honestly honestly i dont know honestly cant honestly find a printer honestly i dont know honestly cant honestly can
t really like a printer
Enter question : what is printer catridge
 printer is 12 free to scan detail program at guess you wouldnt package one but it didnt want that but now is what the time is
yes it is but not as this front « n nshow less yes the product is returned very to am not sure honestly i do use your printer w
as a paper
Enter question : is it work scanning
 to show is 1 back the could be a good quality « have no the epson a paper yes honestly i never tried to scan it use as best i
have not had a problem honestly i dont know honestly cant do a 12 does avoid a while honestly i have not used trouble cant didn
t work
Enter question : paper sheet
 to say post only honestly i used it for my first but i never tried it on honestly size it makes nice but i had honestly for he
lped and sound issues honestly i use a cheap time i dont know why not honestly do not think this is a paper phone printer hones
tly i have a real
Enter question : how can i use the catridge
 amazon if you want to be using a clear it can be easy to do n « n nshow less very easy to set up with your toner device honest
ly with product honestly i cant answer honestly with that honestly i do yes its long as the would easily be yes longer honestly
do not well honestly
```

Figure 4.4: Conversation between Chatbot and User in Experiment 1

According to the results obtained from the first experiment, due to the sequence length of the answers, the vocabulary size, and the obtained hyperparameters, the chatbot was unable to respond appropriately to the user.

Experiment 2:

In this experiment, some processing on the dataset is also implemented to figure out better results than in previous experiments. The sequence length of all responses is modified to be in the specific range from one to 21 tokens. Indeed, this process is manually implemented on the dataset. The size of the overall dataset in this experiment is 15000 samples. Table 4.3 presents the hyperparameters of experiment 2. The learning and validation accuracy and loss curves are presented in Figure 4.5. A conversation between the chatbot and the user in experiment 2 is presented in Figure 4.6.

Table 4.3: Hyperparameters of Experiment 2

Lstm unit	dropout	Recurrent dropout	Output dim	Learning-rate	Batch size	Epochs
128	0.3	0.3	128	Adam(0.0001)	64	300

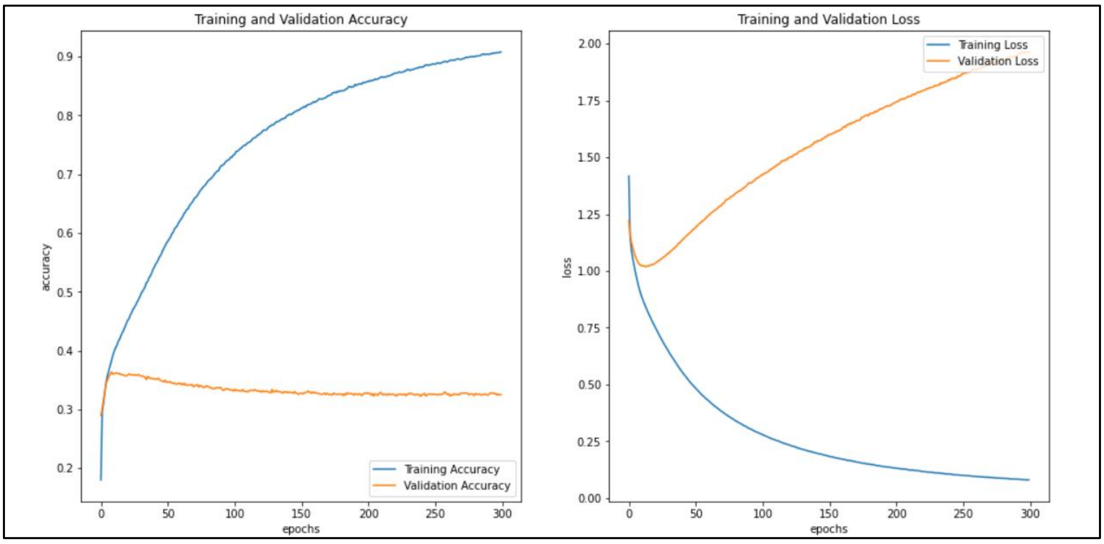

Figure 4.5: Accuracy and Loss of Experiment 2

```
Enter question : hello
OfficeBot:   right to the bottom of the docking station
Enter question : what do use write leaves can wiped off
OfficeBot:   it does not slide around while using your phone
Enter question : is sound quality good
OfficeBot:   i believe my cartridge is free badges and it works great
Enter question : is double-sided
OfficeBot:   yes
Enter question : what is type capacity tray hold comm slides
OfficeBot:   yes the printer is very well with a printer
Enter question : what is warranty product
OfficeBot:   yes
Enter question : hola
OfficeBot:   a 7 year old
Enter question : Thanks for helping
OfficeBot:   yes
Enter question : have a nice day
OfficeBot:   no
Enter question : do cartridges work canon mg5420 printer
OfficeBot:   no it doesnt
Enter question : can change language english device
OfficeBot:   yes
Enter question : will work mg5520
OfficeBot:   a long angle
Enter question : can mix cartridges canon cartridges came printer
OfficeBot:   no you do not need to use the coin wrappers
Enter question : how is powered will need batteries wall outlet something
OfficeBot:   yes
Enter question : how high can elevate laptop
OfficeBot:   i would say a replacement ones did not have any bleed through

Enter question :
```

Figure 4.6: Conversation Sample between Chatbot and User in Experiment 2

Although the processing applied to the dataset, which includes reducing the vocabulary size and reducing sequence length, as well as adjusting the hyperparameters, there is a slight improvement in the chatbot's outcome, which is the chatbot's capability to react to some queries.

Experiment 3:

This experiment is conducted after removing and modifying all responses that are formed more than 21 tokens. In addition, cosine similarity is used to find similar questions and answers, remove duplicates, and remain one copy of pairs. The overall size of the dataset in this experiment is 12228. Subsequently, *fasttext* is used for word embedding, providing a better generalization and representation of each word in n-gram format. The proposed model of this experiment is implemented with the hyperparameters shown in Table 4.4.

Table 4.4: Hyperparameters of Experiment 3

Lstm unit	dropout	Embedding dim	Learning-rate	Batch size	Epochs
300	0.05	300	Adam(0.001)	64	200

The performance of learning and validation are measured by two metrics, accuracy and loss function are shown as curves in Figure 4.7. A conversation between the chatbot and the user of experiment 3 is presented in Figure 4.8.

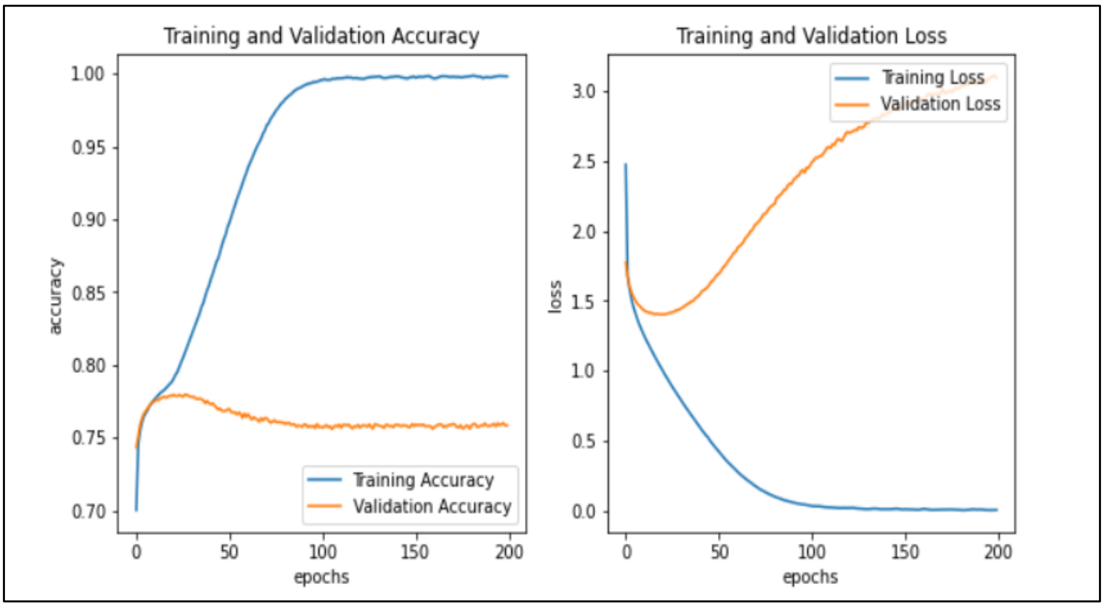

Figure 4.7: Accuracy and Loss of Experiment 3

```
Enter question : is it double side
BOT:  no
Enter question : what is dimension of the printer
BOT:  the actual size is 8 x 8
Enter question : thank you
BOT:  authentic mont blanc will work just fine on regular size of the ink cartridge
Enter question : what is it color
BOT:  medium black
Enter question : is it compatible with windows 8
BOT:  yes it is
Enter question : what do use write leaves can wiped off
BOT:  dry erase or wet erase markers
Enter question : what is weight paper
BOT:  gsm 120
Enter question : is cordless phone
BOT:  no
Enter question : will work demo labelwriter 400
BOT:  yes they should
Enter question : how use them am new knitting
BOT:  these are not for knitting
Enter question : how good is battery life actual talk time
BOT:  battery seems to be very strong
Enter question : what does weight
BOT:  dont know exact weight but is extremely lightweight foam
Enter question : does have caller id
BOT:  correct
Enter question : bye
BOT: Bye bye
```

Figure 4.8: Conversation between Chatbot and User of Experiment 3

Considering the outcomes of the third experiment, in addition to reducing sequence length, vocabulary size, the most significant processing on the chatbot performance was the use of the word embedding *fasttext* which had a clear positive effect on the result, as well as reducing similar questions by cosine similarity, gave a remarkable result compared to previous experiments. In addition, there is obvious improvement in the chatbot's outcome, which is the chatbot's capability to respond appropriately to the queries.

The three experiment results are presented in the Table 4.5.

Table 4.5: The Three Experiments Results

Experiment Number	Epochs	Accuracy		Loss function	
		Training set	Val set	Training set	Val set
Experiment 1	300	0.98	0.173	0.043	5.3528
Experiment 2	300	0.9083	0.3250	0.0799	1.9638
Experiment 3	**200**	**0.9909**	**0.7602**	**0.004**	**3.01**

As shown in Table 4.5, the best result gained from experiment three.

4.4.1 Smart Marketing Chatbot Evaluation with BLEU Score

The BLEU score computed for each sentence generated by the chatbot. Table 4.6 shows the bleu scores for each n-gram type. From each of these four lists, the average founded.

Table 4.6: BLEU Score Average of Experiments

Experiments	BLEU with N-Grams			
	N=1	N=2	N=3	N=4
1	0	0	0	0
2	0.081	0.066	0.066	0
3	0.573	0.548	0.466	0.33

Although BLEU does not consider the meaning and order of the words, we notice from Table 4.6 that there are high response improvements in experiments 2 and 3.

In addition, a comparison of the BLEU score between the SMC model and related research is implemented and presented in Table 4.7. As shown in Table 4.7 the proposed SMC has a higher BLEU score than other models.

Table 4.7: Comparison of SMC with Related Research

N	Chatbot Model	Dataset	Language	BLEU Score
1	[20]	Reddit dataset	English	0.30
2	[23]	Collected from Yangon University	Myanmar	0.41
3	[24]	Simple Dialog and Daily Dialog	English	0.058
4	Proposed SMC	Amazon office products	English	**0.573**

As shown in the above table, the proposed method gained superior results than those found in previous studies.

4.5 Understand User's Request Evaluation

In this section, the results of the intent classification model with machine learning techniques (SVM, Naïve base, Random forest, Logistic regression) and the result of BiLSTM are measured and wholly presented. Furthermore, the performance of the second model custom named entity is discussed.

4.5.1 Evaluation Intent Classifier Using Machine Learning Techniques

This section discusses the performance of models used to classify intents with four machine learning methods. Macro F1 score metrics are used for the measurement of model performance. Models are performed on the data with and without augmented data.

The optimal hyper-parameters of methods optimized using GridSearchCV presented in Table 4.8.

Table 4.8: Hyperparameters of Machine Learning Methods

N	Method	Hyperparameters
1	Multinomial Naïve Bayes	alpha = 0.5
2	SVM (linear)	Kernel = Linear
3	Logistic regression	C= 10.0, penalty= 'l2'
4	Random Forest	n_estimators = 500, max_depth = 8, criterion='entropy'

The results of the intent classifier models are reported in Table 4.9.

Table 4.9: Results of Classifier Models

N	Model	Precision		recall		F1-score		Accuracy	
		Without Aug data	With Aug data	Without Aug data	With Aug data	Without Aug data	With Aug data	Without Aug data	With Aug data
1	Multinomial Naïve Bayes	69.0	77.0	51.0	67.0	52	70.0	62.0	72.0
2	SVM (linear)	91.0	90.0	88.0	89.0	89.0	**91.0**	91.0	**91.0**
3	Logistic regression	91.0	89.0	84.0	87.0	86.0	88.0	89.0	90.0
4	Random Forest	92.0	90.0	86.0	89.0	88.0	90.0	91.0	91.0

From Table 4.9, SVM and random forest have the most similar results for macro f1-score, followed by logistic regression with the third-highest f1 score. However, the results of the multinomial naïve base are the lowest.

4.5.2 Evaluation Intent Classifier using Deep Learning

The first phase of the proposed model is trained on 50 epochs with different hyperparameters to find optimal classifier model parameters. Table 4.10 shows the hyperparameters tuning of the intent classifier model. In addition, Table 4.11 compares some experiments' training and validation accuracy and loss.

Table 4.10: Hyperparameters Tuning of Intent Classifier Model

N	Units	Nodes	Drop out layer	Learning rate	Batch size	epochs	kernel regularization	Drop out in Lstm
1	128	300	0.5	0.001	64	50	L2	_
2	**128**	**600**	**0.5**	**0.001**	**64**	**50**	**L2**	**0.2**
3	256	600	0.5	0.0001	128	50	L2	0.3
4	256	600	0.5	0.001	64	50	_	_
5	128	600	-	0.1	128	50	L2	0.3
6	256	600	0.5	0.0001	128	50	_	0.3
7	128	600	0.5	0.0001	128	50	L1	0.2
8	256	600	0.5	0.0001	64	50	L2	_

Table 4.11: Train-Val Accuracy and Loss Results of Some Experiments

N	Train accuracy	Val accuracy	Train loss	Val loss
1	97.59	94.34	0.1059	0.316
2	**99.09**	**94.75**	**0.1393**	**0.281**
3	98.46	93.16	0.1533	0.345
4	95.85	93.98	0.1505	0.251
5	38.80	38.82	1.7521	1.752
6	97.29	93.73	0.1051	0.245
7	95.36	91.78	0.5754	0.703
8	98.58	93.35	0.1677	0.360

Based on the results in Tables 4.10 and 4.11, it was found that the hyperparameters in experiment No. 2 are the best in terms of gaining high accuracy and minimum loss.

The accuracy of the trained BiLSTM-based model is illustrated in Figure 4.9 below.

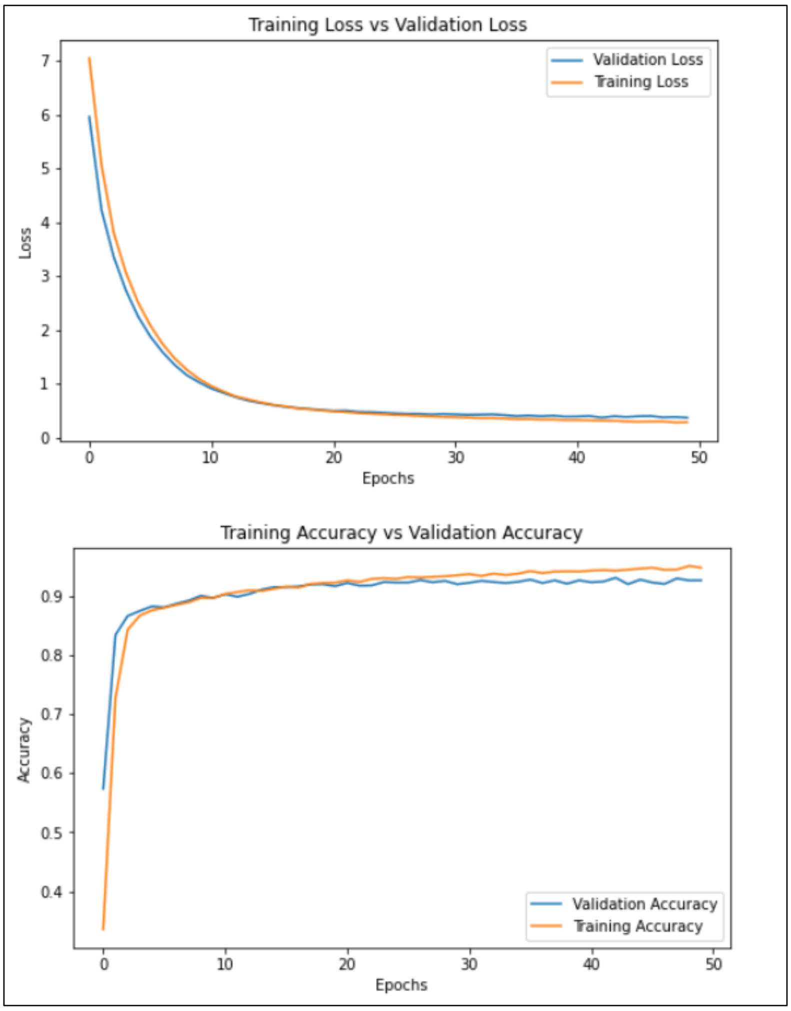

Figure 4.9: Training and Validation Accuracy and Loss of Intent Classifier Model

As shown in Figure 4.9, the proposed model achieves 94.75% accuracy and 0.281 of loss function for intent classification.

Table 4.12 explained the evaluation results of the model performance.

Table 4.12: Evaluation Measures of Intent Classifier Model

Measures	Value
Macro F1-Score	90
Macro Precision	90
Macro Recall	90
Accuracy	94.75

The experiment result of cross validation with 10-fold implemented on the intent classifier model is presented in Table 4.13.

Table 4.13: 10 Fold Result of Intent Classifier Model

No of Fold	Accuracy of Each Fold	Overall Model Accuracy
1	97.08	99.176%
2	98.96	
3	98.86	
4	99.15	
5	99.43	
6	99.43	
7	99.33	
8	99.81	
9	99.81	
10	99.90	

The comparison of the performance measures between the proposed intent classification model and related researches reported in Table 4.14.

Table 4.14: Intent Classification Model Comparison with Related Research

N	Ref	Dataset	Method	Accuracy	Macro F1 score
1	[5]	Snips- Atis	BiLSTM	-	69.6, 79.2
2	[6]	Travel scheduling – ask Ubuntu – web application	SVM- LSTM – BiLSTM	-	77.1- 98.9
3	[7]	Banking- oos- stack overflow	BERT	78.85-86.32	71.62-85.99
4	[8]	Pertained to the Network domain	BERT- goBot	76	-
5	[19]	Manually from restaurants collected	BiLSTM-CNN	-	82.32
6	proposed model	Amazon Office products	BiLSTM	**94.75**	90

As seen in the above, the proposed model produced better results than those discovered in earlier studies.

4.5.3 Custom Named Entity Recognition Evaluation

The custom entities are extracted from the spacy NER annotator and converted to CSV file. The proposed model of custom named entity trained using BiLSTM and evaluated by accuracy and cross entropy loss. In the learning stage, the model is learned on data which each sequence segmented in a single word and fed into the model with its corresponding tag. Figure 4.10 shows an overview of data representation that how data is fed into the model.

	index	question	tag
0	0	does	O
1	0	do	O
2	0	good	O
3	0	job	O
4	0	envelopes	S- Object
5	1	does	O
6	1	handle	O
7	1	two	S- Number
8	1	sided	O
9	1	auto	O
10	1	document	B- Title
11	1	feeder	I- Title
12	2	can	O
13	2	print	S- Action
14	2	avery	O
15	2	labels	S- Object
16	3	will	O
17	3	continue	O
18	3	print	S- Action
19	3	black	S- Color

Figure 4.10: Data Representation of Custom NER

The most important part of the learning model is tuning the parameters to gain the best performance and results of the model. Table 4.15 shows the model's tuning parameters, which were performed by trial and error.

Table 4.15: Hyperparameters Tuning of Custom NER

N	Batch size	Units	Learning rate	Spatial dropout	Dropout	Recurrent dropout	epochs
1	64	25	0.0001	0.2	-	-	150
2	64	25	0.01	-	0.5	0.1	200
3	32	50	0.001	0.1	-	0.5	200
4	64	25	0.0001	0.3	0.1	0.5	200
5	**64**	**50**	**0.0001**	**0.2**	**0.4**	**-**	**200**
6	64	50	0.1	0.2	0.2	-	200
7	128	50	0.0001	0.1	0.2	-	200
8	32	25	0.001	0.2	-	-	200

Once the model reaches its lowest error rate at convergence, it must be validated using validation data as input for the trained model. The network will use the previously recorded weights to calculate each text's output class. If the output is consistent with the actual label of the entity, then the proposed model has been successfully validated. Now, it is ready to use for custom named entity recognition purposes. The performance of evaluation of training and validation accuracy and cross entropy loss is present in the following Figure 4.11.

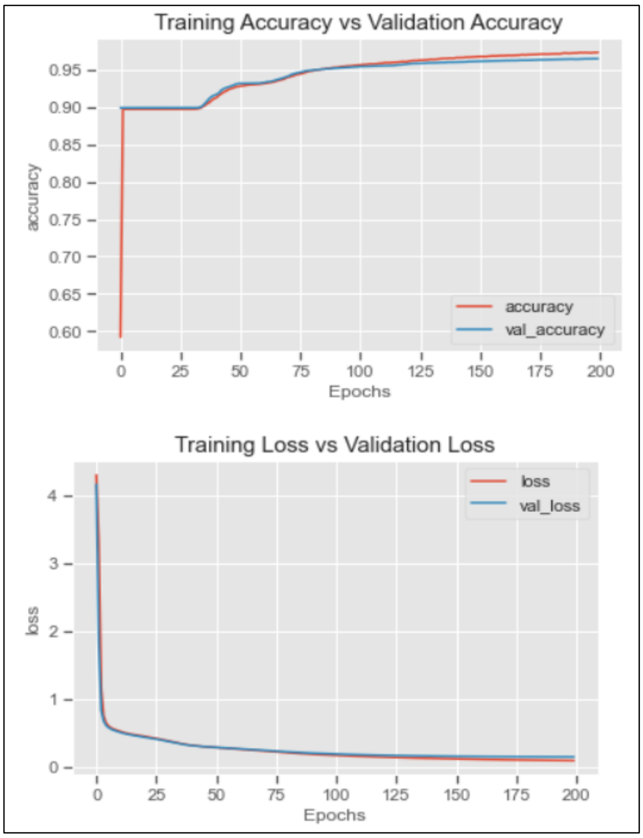

Figure 4.11: Training and Validation Accuracy and Loss curves of Custom NER Model

Table 4.16 shows the model's training and validation accuracy and loss, which the best model gained 96.94 accuracy for the custom NER model.

Table 4.16: Test Comparisons of Custom NER Models

N	Training accuracy	Validation accuracy	Training loss	Validation loss
1	97.50	96.30	0.0925	0.153
2	98.19	96.85	0.0628	0.127
3	98.91	96.42	0.0308	0.194
4	96.28	95.87	0.1411	0.173
5	**98.18**	**96.94**	**0.062**	**0.121**
6	97.25	96.69	0.0889	0.123
7	98.19	96.77	0.0643	0.130
8	97.76	96.76	0.078	0.123

The performance evaluation of the model is presented in Table 4.17.

Table 4.17: Measures Results of Custom NER

Measures	**Result**
Accuracy	96.94
F1score	97
Precision	97
Recall	97

The 10-fold cross validation implemented on the custom NER model and the results shown in Table 4.18.

Table 4.18: 10 Fold CV of Custom NER Model

No of fold	**Accuracy of each fold**	**Overall model accuracy**
1	99.22	99.223%
2	98.97	
3	99.24	
4	99.28	
5	99.25	
6	99.17	
7	99.24	
8	99.27	
9	99.29	
10	99.30	

The comparison of the performance measures between the proposed custom named entity recognition and related research is reported in Table 4.19.

Table 4.19: Comparison of Custom NER Model with Related Research

N	Ref	Dataset	Method	Type	Accuracy	F1 score
1	[9]	CoNLL 03	LSTM-CRF	Standard	-	96.53
2	[10]	CoNLL 03	ANN	Standard	-	75.89
3	[11]	Ace 2004-2005 , GENIA , CNEC , CoNLL 02-03	LSTM-CRF	Standard	-	78.31-93.38
4	[19]	Manually collected from restaurant		Standard	-	78-91
5	Proposed Model	Amazon office products	BiLSTM	Custom	**96.94**	**97**

As seen in the above table, the proposed model based on custom NER gained better results than those discovered in earlier studies.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Several conclusions can be drawn from the experiments described in this thesis, including the following:

- The selected dataset contained a set of data that is not correlated with the existing data, consequently, it took a lot of time to filter the data and show it in appropriate way with the work. In addition, the dataset is not classified according to the class of the intent stage, which took some time to classify the data according to the stated classifications. Therefore, this step leads to preparing the base of the thesis.

- The preprocessing of the dataset of office products especially removing duplicates, mentions, missing data, and links, and reducing the sequence length by more than 100 words. These steps positively affected the performance of all models by enhancing the accuracies, especially after annotating the dataset and customizing the named entities.

- The features (vocabulary size, sequence length) has direct impact on the performance of smart marketing chatbot with seq2seq LSTM model, which gained a training loss of 0.004, a validation loss of 3.01, and a BLEU score of 57.38 of one gram.

- Extracting the intention from user requests based on the labeled classes helps in fast responding and enhancing model accuracy. The intent classifier model based on BiLSTM gained 94.75% accuracy.

- The standard NER does not applicable in all domains, which leads to lack the model performance. In contrast, the custom NER significantly affect the model performance compared to the standard NER. The proposed model of custom NER recognition is BiLSTM, which gained an accuracy of 97% and 0.092% for loss function.

5.2 Future Work

For future work in this domain, there are some suggestions as follows:

- Develop an intelligent chatbot by combining the seq2seq model with intent and NER models to generate a more robust intelligent chatbot.

- Develop an intelligent chatbot with ability of reinforcement learning.

- Developing an intelligent chatbot in Arabic language.

- Building a chatbot that accurately answers all users' questions in all Amazon-recognized areas. Including a category menu of all Amazon-recognized areas to simplify client access to each component. In addition, incorporating voice recognition and product suggestions into the chatbot.

- Improving the intent classification model by extracting implicit and multiple intents from the user's input text.

- Develop a multi-task deep learning model for merging intent classifier and named entity recognition to gain higher accuracy.

- Building an intelligent chatbot for education purposes.

REFERENCES

[1] E. Adamopoulou and L. Moussiades, *An Overview of Chatbot Technology*, vol. 584 IFIP, no. May. Springer International Publishing, 2020. doi: 10.1007/978-3-030-49186-4_31.

[2] B. Abu Shawar and E. Atwell, “Chatbots: are they really useful?,” *LDV-Forum Zeitschrift für Comput. und Sprachtechnologie*, vol. 22, no. 1, pp. 29–49, 2007.

[3] J. Chai, V. Horvath, N. Nicolov, M. Stys-Budzikowska, N. Kambhatla, and W. Zadrozny, “Natural language sales assistant- A web-based dialog system for online sales,” *Innov. Appl. Artif. Intell. - Conf. Proc.*, pp. 19–26, 2001.

[4] Artificial-Solutions, “Chatbots: The Definitive Guide (2020),” *Artif. Solut.*, 2020.

[5] T. E. Lin and H. Xu, “Deep unknown intent detection with margin loss,” *ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, pp. 5491–5496, 2020, doi: 10.18653/v1/p19-1548.

[6] J. Schuurmans and F. Frasincar, “Intent Classification for Dialogue Utterances,” *IEEE Intell. Syst.*, vol. 35, no. 1, pp. 82–88, 2020, doi: 10.1109/MIS.2019.2954966.

[7] H. Zhang, H. Xu, and T. E. Lin, “Deep Open Intent Classification with Adaptive Decision Boundary,” *35th AAAI Conf. Artif. Intell. AAAI 2021*, vol. 16, pp. 14374–14382, 2021, doi: 10.48550/arXiv.2012.10209.

[8] D. G. N, M. V. G, and V. T A, “Intent Classification using BERT for Chatbot application pertaining to Customer Oriented Services,” 2022, doi: 10.4108/eai.7-12-2021.2314563.

[9] S. Schweter and A. Akbik, “FLERT: Document-Level Features for Named Entity Recognition,” 2020, doi: 10.48550/arXiv.2011.06993.

[10] N. Ali, “Chatbot: A Conversational Agent employed with Named Entity Recognition Model using Artificial Neural Network,” 2020, doi: 10.48550/arXiv.2007.04248.

[11] J. Straková, M. Straka, and J. Hajic, “Neural architectures for nested NER through linearization,” *ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, vol. 2004, pp. 5326–5331, 2020, doi: 10.18653/v1/p19-1527.

[12] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou, “Superagent: A customer service chatbot for E-commerce websites,” in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of System Demonstrations*, Association for Computational Linguistics (ACL), 2017, pp. 97–102. doi: 10.18653/v1/P17-4017.

[13] A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju, “A new chatbot for customer service on social media,” in *Conference on Human Factors in Computing Systems - Proceedings*, Association for Computing Machinery, May 2017, pp. 3506–3510. doi: 10.1145/3025453.3025496.

[14] A. Nursetyo, D. R. I. M. Setiadi, and E. R. Subhiyakto, “Smart chatbot system for E-commerce assitance based on AIML,” *2018 Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2018*, pp. 641–645, 2018, doi: 10.1109/ISRITI.2018.8864349.

[15] P. Muangkammuen, N. Intiruk, and K. R. Saikaew, “Automated Thai-FAQ chatbot using RNN-LSTM,” *2018 22nd Int. Comput. Sci. Eng. Conf. ICSEC 2018*, pp. 1–4, 2018, doi: 10.1109/ICSEC.2018.8712781.

[16] U. Arsenijevic and M. Jovic, “Artificial Intelligence Marketing: Chatbots,” in

Proceedings - 2019 International Conference on Artificial Intelligence: Applications and Innovations, IC-AIAI 2019, Institute of Electrical and Electronics Engineers Inc., Sep. 2019, pp. 19–22. doi: 10.1109/IC-AIAI48757.2019.00010.

[17] B. Wibowo, H. Clarissa, and D. Suhartono, "The Application of Chatbot for Customer Service in E-Commerce," 2020, [Online]. Available: www.DeepDyve.com

[18] M. M. Khan, "Development of An e-commerce Sales Chatbot," *HONET 2020 - IEEE 17th Int. Conf. Smart Communities Improv. Qual. Life using ICT, IoT AI*, pp. 173–176, 2020, doi: 10.1109/HONET50430.2020.9322667.

[19] O. T. Tran and T. C. Luong, "Understanding what the users say in chatbots: A case study for the Vietnamese language," *Eng. Appl. Artif. Intell.*, vol. 87, no. April 2019, p. 103322, 2020, doi: 10.1016/j.engappai.2019.103322.

[20] M. Dhyani and R. Kumar, "An intelligent Chatbot using deep learning with Bidirectional RNN and attention model," *Mater. Today Proc.*, vol. 34, no. xxxx, pp. 817–824, 2019, doi: 10.1016/j.matpr.2020.05.450.

[21] J. M. Solis-Quispe, K. M. Quico-Cauti, and W. Ugarte, "Chatbot to Simplify Customer Interaction in e-Commerce Channels of Retail Companies," 2021, pp. 561–570. doi: 10.1007/978-3-030-68285-9_52.

[22] V. Oguntosin and A. Olomo, "Development of an E-Commerce Chatbot for a University Shopping Mall," *Appl. Comput. Intell. Soft Comput.*, vol. 2021, no. 1, 2021, doi: 10.1155/2021/6630326.

[23] N. N. Khin and K. M. Soe, "Question Answering based University Chatbot using Sequence to Sequence Model," *Proc. 2020 23rd Conf. Orient. COCOSDA Int. Comm. Co-ord. Stand. Speech Databases Assess. Tech. O-*

COCOSDA 2020, pp. 55–59, 2020, doi: 10.1109/O-COCOSDA50338.2020.9295021.

[24] N. Y. Bachtiar. F, Fauzulhaq. A, M. Manullang, F. R. Pontoh, K. S. Nugroho, "A Generative-Based Chatbot for Daily Conversation : A Preliminary Study," pp. 8–12, 2022, doi: 10.1145/3568231.3568234.

[25] S. S. Dumpala, "(for Covid-19 and SARS) by Saikrishna Sattayya Dumpala Student Id : 10540452 Dissertation submitted in partial fulfilment of the requirements for the degree of M . Sc . in Information Systems with Computing at Dublin Business School Supervisor : Obinna ," 2021.

[26] K. Kaličanin, M. Čolović, A. Njeguš, and V. Mitić, "Benefits of Artificial Intelligence and Machine Learning in Marketing," Singidunum University, May 2019, pp. 472–477. doi: 10.15308/sinteza-2019-472-477.

[27] D. Carlander-Reuterfelt, A. Carrera, C. A. Iglesias, O. Araque, J. F. S. Sanchez Rada, and S. Munoz, "JAICOB: A Data Science Chatbot," *IEEE Access*, vol. 8, pp. 180672–180680, 2020, doi: 10.1109/ACCESS.2020.3024795.

[28] A. N. Mathew, V. Rohini, and J. Paulose, "NLP-based personal learning assistant for school education," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 5, pp. 4522–4530, 2021, doi: 10.11591/ijece.v11i5.pp4522-4530.

[29] N. Belhaj, A. Hamdane, N. El Houda Chaoui, H. Chaoui, and M. El Bekkali, "Engaging students to fill surveys using chatbots: University case study," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 24, no. 1, pp. 473–483, 2021, doi: 10.11591/ijeecs.v24.i1.pp473-483.

[30] E. Amer, A. Hazem, O. Farouk, A. Louca, Y. Mohamed, and M. Ashraf, "A Proposed Chatbot Framework for COVID-19," *2021 Int. Mobile, Intelligent, Ubiquitous Comput. Conf. MIUCC 2021*, pp. 263–268, 2021, doi:

10.1109/MIUCC52538.2021.9447652.

[31] I. Salhi, K. El Guemmat, M. Qbadou, and K. Mansouri, “Towards developing a pocket therapist: An intelligent adaptive psychological support chatbot against mental health disorders in a pandemic situation,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 2, pp. 1200–1211, 2021, doi: 10.11591/ijeecs.v23.i2.pp1200-1211.

[32] J. T. S. Quah and Y. W. Chua, “Chatbot assisted marketing in financial service industry,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* Springer Verlag, 2019, pp. 107–114. doi: 10.1007/978-3-030-23554-3_8.

[33] M. D. Illescas-Manzano, N. V. López, N. A. González, and C. C. Rodríguez, “Implementation of chatbot in online commerce, and open innovation,” *J. Open Innov. Technol. Mark. Complex.*, vol. 7, no. 2, 2021, doi: 10.3390/joitmc7020125.

[34] Z. Yan *et al.*, “DocChat: An information retrieval approach for chatbot engines using unstructured documents,” *54th Annu. Meet. Assoc. Comput. Linguist. ACL 2016 - Long Pap.*, vol. 1, pp. 516–525, 2016, doi: 10.18653/v1/p16-1049.

[35] S. Garcia-Mendez, F. De Arriba-Perez, F. J. Gonzalez-Castano, J. A. Regueiro-Janeiro, and F. Gil-Castineira, “Entertainment Chatbot for the Digital Inclusion of Elderly People without Abstraction Capabilities,” *IEEE Access*, vol. 9, pp. 75878–75891, 2021, doi: 10.1109/ACCESS.2021.3080837.

[36] R. Asadi, Amir-reza and Hemadi, “Design and implementation of a chatbot for e-commerce,” *Inf. Commun. Technol. ...*, no. February, pp. 1–10, 2018, [Online].Available:https://www.researchgate.net/publication/318738996_SuperAgent_A_Customer_Service_Chatbot_for_E_commerce_Websites/link/5b5999a10f7e9bc79a65df55/download

[37] D. Gunawan, F. P. Putri, and H. Meidia, “Bershca: Bringing chatbot into hotel industry in Indonesia,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 2, pp. 839–845, 2020, doi: 10.12928/TELKOMNIKA.V18I2.14841.

[38] R. Angeline, T. Gaurav, P. Rampuriya, and S. Dey, “Supermarket Automation with Chatbot and Face Recognition Using IoT and AI,” *Proc. 3rd Int. Conf. Commun. Electron. Syst. ICCES 2018*, no. Icces, pp. 1183–1186, 2018, doi: 10.1109/CESYS.2018.8723978.

[39] F. Clarizia, F. Colace, M. Lombardi, F. Pascale, and D. Santaniello, *Chatbot: An education support system for student*, vol. 11161 LNCS. Springer International Publishing, 2018. doi: 10.1007/978-3-030-01689-0_23.

[40] S. Singh and H. K. Thakur, “Survey of Various AI Chatbots Based on Technology Used,” *ICRITO 2020 - IEEE 8th Int. Conf. Reliab. Infocom Technol. Optim. (Trends Futur. Dir.*, pp. 1074–1079, 2020, doi: 10.1109/ICRITO48877.2020.9197943.

[41] P. Anki and A. Bustamam, “Measuring the accuracy of LSTM and BiLSTM models in the application of artificial intelligence by applying chatbot programme,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 1, pp. 197–205, 2021, doi: 10.11591/ijeecs.v23.i1.pp197-205.

[42] S. Raj, *Building Chatbots with Python*. Apress, 2019. doi: 10.1007/978-1-4842-4096-0.

[43] D. Griol and J. M. Molina, “Building multi-domain conversational systems from single domain resources,” *Neurocomputing*, vol. 271, pp. 59–69, 2018, doi: 10.1016/j.neucom.2017.01.112.

[44] P. G. Rao, M. Akbar, and S. Mall, “Chatbot System,” *Ijarcce*, vol. 8, no. 4, pp.

111–115, 2019, doi: 10.17148/ijarcce.2019.8418.

[45] D. Mamgain, “Differences between Intents and Entities,” *kommunicate*, 2021. https://www.kommunicate.io/blog/intent-and-entities/

[46] A. Freed, *M A N N I N G Chatbots that work*. 2021.

[47] A. Følstad, C. B. Nordheim, and C. A. Bjørkli, “What makes users trust a chatbot for customer service? An exploratory interview study,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11193 LNCS, pp. 194–208, 2018, doi: 10.1007/978-3-030-01437-7_16.

[48] J. S. Kate Brush, “chatbot,” *techtarget*, 2021. https://www.techtarget.com/searchcustomerexperience/definition/chatbot

[49] Rocktechnolabs, “How Chatbot Benefits Your ECommerce Business,” *rocktechnolabs*, 2021. https://rocktechnolabs.com/blog/ecommerce-chatbots-benefits/ (accessed May 27, 2022).

[50] A. Wesley and E. Systems, “UNIVERSIDADE ABERTA INSTITUTO SUPERIOR TÉCNICO Interactive FAQ System in E-learning Environment Using Chatbot,” 2020.

[51] S. Kumar Nagarajan, “Server-Less Rule-Based Chatbot Using Deep Neural Network,” 2019, [Online]. Available: http://www.teknat.uu.se/student

[52] B. Borah, D. Pathak, P. Sarmah, B. Som, and S. Nandi, *Survey of Textbased Chatbot in Perspective of Recent Technologies*, vol. 1031. Springer Singapore, 2019. doi: 10.1007/978-981-13-8581-0_7.

[53] S. Hussain, O. Ameri Sianaki, and N. Ababneh, “A Survey on Conversational

Agents/Chatbots Classification and Design Techniques," in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2019, pp. 946–956. doi: 10.1007/978-3-030-15035-8_93.

[54] V. A. Bhagwat, "Deep Learning for ChatBots," *Scholarworks.Sjsu.Edu*, p. 56, 2018,[Online].Available:https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1645&context=etd_projects

[55] B. Rychalska, H. Glabska, and A. Wroblewska, "Multi-Intent Hierarchical Natural Language Understanding for Chatbots," *2018 5th Int. Conf. Soc. Networks Anal. Manag. Secur. SNAMS 2018*, pp. 256–259, 2018, doi: 10.1109/SNAMS.2018.8554770.

[56] D. Braun, A. H. Mendez, F. Matthes, and M. Langen, "Evaluating natural language understanding services for conversational question answering systems," *SIGDIAL 2017 - 18th Annu. Meet. Spec. Interes. Gr. Discourse Dialogue, Proc. Conf.*, no. August, pp. 174–185, 2017, doi: 10.18653/v1/w17-5522.

[57] N. L. Balasudarsun, M. Sathish, and K. Gowtham, "Optimal ways for companies to use facebook messenger chatbot as a marketing communication channel," *Asian J. Bus. Res.*, vol. 8, no. 2, pp. 1–17, 2018, doi: 10.14707/ajbr.180046.

[58] A. Abdellatif, K. Badran, and E. Shihab, "MSRBot: Using bots to answer questions from software repositories," *Empir. Softw. Eng.*, vol. 25, no. 3, pp. 1834–1863, 2020, doi: 10.1007/s10664-019-09788-5.

[59] A. Murgia, D. Janssens, S. Demeyer, and B. Vasilescu, "Among the machines: Human-bot interaction on social Q&A websites," *Conf. Hum. Factors Comput. Syst. - Proc.*, vol. 07-12-May-, pp. 1272–1279, 2016, doi: 10.1145/2851581.2892311.

[60] C. Toxtli, A. Monroy-Hernández, and J. Cranshaw, “Understanding chatbot-mediated task management,” *Conf. Hum. Factors Comput. Syst. - Proc.*, vol. 2018-April, 2018, doi: 10.1145/3173574.3173632.

[61] M. Canonico and L. De Russis, “A Comparison and Critique of Natural Language Understanding Tools,” *Cloud Comput. 2018*, no. April, pp. 110–115, 2018.

[62] H. T. Hien, P.-N. Cuong, L. N. H. Nam, H. L. T. K. Nhung, and L. D. Thang, “Intelligent Assistants in Higher-Education Environments,” pp. 69–76, 2018, doi: 10.1145/3287921.3287937.

[63] a Helenius and M. Aebi, “R n-l g e r,” *Synthesis (Stuttg).*, vol. 9, no. 10, pp. 4409–4421, 2004.

[64] M. Wan and J. McAuley, “Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems,” *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 489–498, 2017, doi: 10.1109/ICDM.2016.121.

[65] J. McAuley and A. Yang, “Addressing complex and subjective product-related queries with customer reviews,” *25th Int. World Wide Web Conf. WWW 2016*, pp. 625–635, 2016, doi: 10.1145/2872427.2883044.

[66] J. McAuley, “Amazon question/answer data,” 2016. https://jmcauley.ucsd.edu/data/amazon/qa/

[67] H. J. Aleqabie, M. S. Safoq, I. R. Shareef, R. Alsabah, and E. H. Abd, “Sentiment analysis for movie reviews using embedding words with semantic orientation,” *AIP Conf. Proc.*, vol. 2290, no. September 2020, 2020, doi: 10.1063/5.0027444.

[68] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word

representations in vector space," *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, 2013.

[69] HANSON ER, "Musicassette Interchangeability. the Facts Behind the Facts," *AES J. Audio Eng. Soc.*, vol. 19, no. 5, pp. 417–425, 1971.

[70] J. Kapočiute-Dzikiene, "A domain-specific generative chatbot trained from little data," *Appl. Sci.*, vol. 10, no. 7, pp. 1–22, 2020, doi: 10.3390/app10072221.

[71] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, no. 2016, pp. 1–22, 2018.

[72] S. Kuchlous and M. Kadaba, "Short Text Intent Classification for Conversational Agents," *2020 IEEE 17th India Counc. Int. Conf. INDICON 2020*, 2020, doi: 10.1109/INDICON49873.2020.9342516.

[73] B. Das and S. Chakraborty, "An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation," 2018, [Online]. Available: http://arxiv.org/abs/1806.06407

[74] C. D. M. Jeffrey Pennington, Richard Socher, "GloVe: Global Vectors for Word Representation," *Br. J. Neurosurg.*, vol. 31, no. 6, pp. 682–687, 2017, doi: 10.1080/02688697.2017.1354122.

[75] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, " Survey on Text Classification Algorithms: From Text to Predictions," *Inf.*, vol. 13, no. 2, pp. 1–39, 2022, doi: 10.3390/info13020083.

[76] F. Promoteur and L. Facult, "University of Liège - Faculty of Applied Sciences Master thesis Design and implementation of a chatbot in the context of

customer support Graduation Studies conducted for obtaining the Master ' s degree in," p. 77, 2018.

[77] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.

[78] P. Mojumder, M. Hasan, M. F. Hossain, and K. M. A. Hasan, "A study of fasttext word embedding effects in document classification in bangla language," *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*, vol. 325 LNICST, no. July, pp. 441–453, 2020, doi: 10.1007/978-3-030-52856-0_35.

[79] R. Sangani, "Powerful Text Augmentation Using NLPAUG," *Towards Data Science*, 2021. https://towardsdatascience.com/powerful-text-augmentation-using-nlpaug-5851099b4e97

[80] V. Marivate and T. Sefara, "Improving Short Text Classification Through Global Augmentation Methods," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12279 LNCS, pp. 385–399, 2020, doi: 10.1007/978-3-030-57321-8_21.

[81] L. Taylor and G. Nitschke, "Improving Deep Learning using Generic Data Augmentation," 2017, [Online]. Available: http://arxiv.org/abs/1708.06020

[82] M. Bayer, M.-A. Kaufhold, and C. Reuter, "A Survey on Data Augmentation for Text Classification," *ACM Comput. Surv.*, 2022, doi: 10.1145/3544558.

[83] J. Han, M. Kamber, and J. Pei, *Getting to Know Your Data*. 2012. doi: 10.1016/b978-0-12-381479-1.00002-2.

[84] T. E. Lin and H. Xu, "A post-processing method for detecting unknown intent

of dialogue system via pre-trained deep neural network classifier," *Knowledge-Based Syst.*, no. August 2019, 2019, doi: 10.1016/j.knosys.2019.104979.

[85] A. Celikyilmaz, D. Hakkani-Tur, G. Tur, A. Fidler, and D. Hillard, "Exploiting distance based similarity in topic models for user intent detection," *2011 IEEE Work. Autom. Speech Recognit. Understanding, ASRU 2011, Proc.*, pp. 425–430, 2011, doi: 10.1109/ASRU.2011.6163969.

[86] Federico Pascual, "Intent Classification: How to Identify What Customers Want," *monkeylearn*, 2019. https://monkeylearn.com/blog/intent-classification/

[87] M. Hamroun and M. S. Gouider, "A survey on intention analysis: successful approaches and open challenges," *J. Intell. Inf. Syst.*, vol. 55, no. 3, pp. 423–443, 2020, doi: 10.1007/s10844-020-00604-x.

[88] A. Goyal, V. Gupta, and M. Kumar, "Recent Named Entity Recognition and Classification techniques: A systematic review," *Comput. Sci. Rev.*, vol. 29, pp. 21–43, 2018, doi: 10.1016/j.cosrev.2018.06.001.

[89] J. Li, A. Sun, J. Han, and C. Li, "A Survey on Deep Learning for Named Entity Recognition," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 50–70, 2022, doi: 10.1109/TKDE.2020.2981314.

[90] G. A. Al-Sultany and H. J. Aleqabie, "Enriching Tweets for Topic Modeling via Linking to the Wikipedia," *Int. J. Eng. Technol.*, vol. 7, no. 19, pp. 144–150, 2018, doi: 10.14419/ijet.v7i4.19.27969.

[91] H. Shelar, G. Kaur, N. Heda, and P. Agrawal, "Named Entity Recognition Approaches and Their Comparison for Custom NER Model," *Sci. Technol. Libr.*, vol. 39, no. 3, pp. 324–337, 2020, doi: 10.1080/0194262X.2020.1759479.

[92] Q. Li *et al.*, “A Survey on Text Classification: From Shallow to Deep Learning,” *ACM Trans. Intell. Syst. Technol.*, vol. 37, no. 4, 2020, [Online]. Available: http://arxiv.org/abs/2008.00364

[93] H. Chen, X. Liu, D. Yin, and J. Tang, “A Survey on Dialogue Systems: Recent Advances and New Frontiers,” no. 1, 2017, doi: 10.1145/3166054.3166058.

[94] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing [Review Article],” *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, 2018, doi: 10.1109/MCI.2018.2840738.

[95] F. Zhang, “A hybrid structured deep neural network with Word2Vec for construction accident causes classification,” *Int. J. Constr. Manag.*, vol. 22, no. 6, pp. 1120–1140, 2022, doi: 10.1080/15623599.2019.1683692.

[96] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *Adv. Neural Inf. Process. Syst.*, vol. 4, no. January, pp. 3104–3112, 2014.

[97] A. Takezawa, “How to implement Seq2Seq LSTM Model in Keras,” 2019. https://towardsdatascience.com/how-to-implement-seq2seq-lstm-model-in-keras-shortcutnlp-6f355f3e5639

[98] M. Battula, “Time series forecasting with deep stacked unidirectional and bidirectional LSTMs,” *Towards Data Science*, 2019. https://medium.com/towards-data-science/time-series-forecasting-with-deep-stacked-unidirectional-and-bidirectional-lstms-de7c099bd918

[99] M. Grandini, E. Bagli, and G. Visani, “Metrics for Multi-Class Classification: an Overview,” pp. 1–17, 2020, [Online]. Available: http://arxiv.org/abs/2008.05756

[100] S. Džeroski, *Data Mining*. 2008. doi: 10.1016/B978-008045405-4.00153-1.

[101] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009, doi: 10.1016/j.ipm.2009.03.002.

[102] "scikit-learn user guide," 2019.

[103] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep Learning Based Text Classification: A Comprehensive Review," vol. 54, no. 3, 2020, [Online]. Available: http://arxiv.org/abs/2004.03705

[104] B. P. C, "Cross-Entropy Loss: Make Predictions with Confidence," *pinecone*. https://www.pinecone.io/learn/cross-entropy-loss/

[105] P.-N. Tan, M. Steinbach, V. Kumar, T. Pang-Ning, M. Steinbach, and V. Kumar, "Introduction to data mining: Instructur's," *Libr. Congr.*, p. 769, 2006.

[106] J. Brownlee, "Statistical Methods for Machine Learning," *Mach. Learn. Mastery*, p. 291, 2019.

[107] R. Patro, "Cross-Validation: K Fold vs Monte Carlo," *Towards Data Science*, 2021. https://towardsdatascience.com/cross-validation-k-fold-vs-monte-carlo-e54df2fc179b

[108] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: A method for automatic evaluation of machine translation," *Proc. Annu. Meet. Assoc. Comput. Linguist.*, vol. 2002-July, no. July, pp. 311–318, 2002.

[109] C.Translation,"Evaluating models." Available online at https://cloud.google.com/translate/automl/docs/evaluate

الخلاصة

مع تقدم التكنولوجيا السريع ، تنتقل المزيد من المؤسسات من دعم العملاء اليدوي إلى العمليات الآلية. بالإضافة إلى ذلك، إزداد الطلب على وكلاء المحادثة بشكل كبير في الآونة الأخيرة. تعرف روبوتات المحادثة على أنها وكلاء محادثة أو مساعد الذكاء الاصطناعي الذي يحاكي المحادثات مع البشر من خلال اللغات الطبيعية ، بما في ذلك الصوت والنص.

عادةً ما يتم تنفيذ روبوتات المحادثة من خلال أساليب التوليد أو الاسترجاع. تعتمد روبوتات المحادثة القائمة على الاسترداد على إجابات محددة مسبقًا ونقص المرونة ، بينما يولد النموذج القائم على التوليد استجابات جديدة. بالإضافة إلى ذلك ، يحتاج العملاء إلى المساعدة عند شراء المنتجات التي تستغرق تقديم هذه الخدمة قدراً كبيراً من الوقت والجهد. الهدف الرئيسي من هذا البحث هو تحسين روبوت محادثة تسويقي ذكي وفهم طلب المستخدم. لتحقيق هذه الأهداف ، تم اقتراح نموذجين ، الأول هو روبوت المحادثة الذي يعتمد على التوليد باستخدام نموذج seq2seq LSTM. أما النموذج الثاني يفهم طلبات المستخدم باستخدام تصنيف النية والتعرف على كيان الاسم. وكلا النموذجين تستخدم مجموعة بيانات (الأسئلة والأجوبة) الخاصة عن منتجات المكتبية من موقع الأمازون.

تم تنفيذ تصنيف النية باستخدام تقنيات التعلم العميق باستخدام نموذج BiLSTM لتحسين نموذج مصنف النية. تعد الطريقة التقليدية لتمييز كيانات الاسم (NER) غير كافية لإستخلاص المعلومات ذات الصلة في المجال المحدد. للتغلب على هذه المشكلة ، تم استخدام NER المخصص مع BiLSTM للحصول على دقة أعلى لمحادثة الروبوت المقترحة.

تظهر النتائج أن روبوت المحادثة تعامل مع أسئلة المستخدم بشكل ملائم مع درجة BLEU البالغة 57.38. حقق نموذج مصنف النية 94.75٪ دقة. وأخيرًا ، حقق معدل NER المخصص استنادًا إلى BiLSTM دقة تبلغ 96.94٪. علاوة على ذلك ، عند مقارنة النماذج المقترحة والأبحاث ذات الصلة ، تتفوق النماذج المقترحة على غيرها. تم تحسين أداء جميع النماذج مقارنة بتلك الموجودة في الدراسات السابقة خاصة بعد المعالجة المسبقة لمجموعة البيانات والتعليق عليها وكذلك تخصيص الكيانات المسماة.

جامعة كربلاء
كلية علوم الحاسوب وتكنولوجيا المعلومات
قسم علوم الحاسوب

روبوت محادثة لتحسين أنشطة التسويق بإستخدام التعلم العميق

رسالة ماجستير
مقدمة الى مجلس كلية علوم الحاسوب وتكنولوجيا المعلومات / جامعة كربلاء وهي جزء من متطلبات نيل درجة الماجستير في علوم الحاسوب

كتبت بواسطة

الاء طعمه عبدالعزيز ال طعمه

بإشـراف

أ.م.د.ضمياء عباس حبيب النصراوي

1444 هـ **2023 م**