NATURAL LANGUAGE PROCESSING FOR QUESTION ANSWERING IN LEGAL DOMAIN

A PROJECT REPORT

Submitted by

AMAN SAINI (19BCS1733)

ABHINAV VERMA (19BCS1232)

ASHISH (19BCS1964)

RUPAM BHANDARI (19BCS1840)

ATUL KUMAR SHARMA (19BCS3075)

in partial fulfillment of the requirements for the award of degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Chandigarh University

MAY 2023

BONAFIDE CERTIFICATE

Certified that this project report "**NATURAL LANGUAGE PROCESSING FOR QUESTION ANSWERING IN LEGAL DOMAIN**" is the bonafide work of "**AMAN SAINI (19BCS1733), RUPAM BHANDARI (19BCS1840), ABHINAV VERMA (19BCS1232), ATUL KUMAR SHARMA (19BCS3075), ASHISH (19BCS1964)**", who carried out the project work under my/our supervision.

SIGNATURE **SIGNATURE**

Dr. Navpreet Kaur Walia Inakshi Garg

HEAD OF THE DEPARTMENT **SUPERVISOR**

Computer Science & Engineering Computer Science & Engineering

Submitted for the project viva-voce examination held on_

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We, students of Bachelor of Engineering in Computer Science & Engineering, 8th Semester, session: Feb-May 2023, Chandigarh University, hereby declare that the work presented in this Project Report entitled "Natural Language Processing for Question Answering in Legal Domain" is the outcome of ourown work, is bonafide and correct to the best of our knowledge and this work hasbeen carried out taking care of Engineering Ethics. The work presented does notinfringe any patented work and has not beensubmitted to any other university oranywhere else for the award of any degree or any professional diploma.

TABLE OF CONTENTS

List of Figures

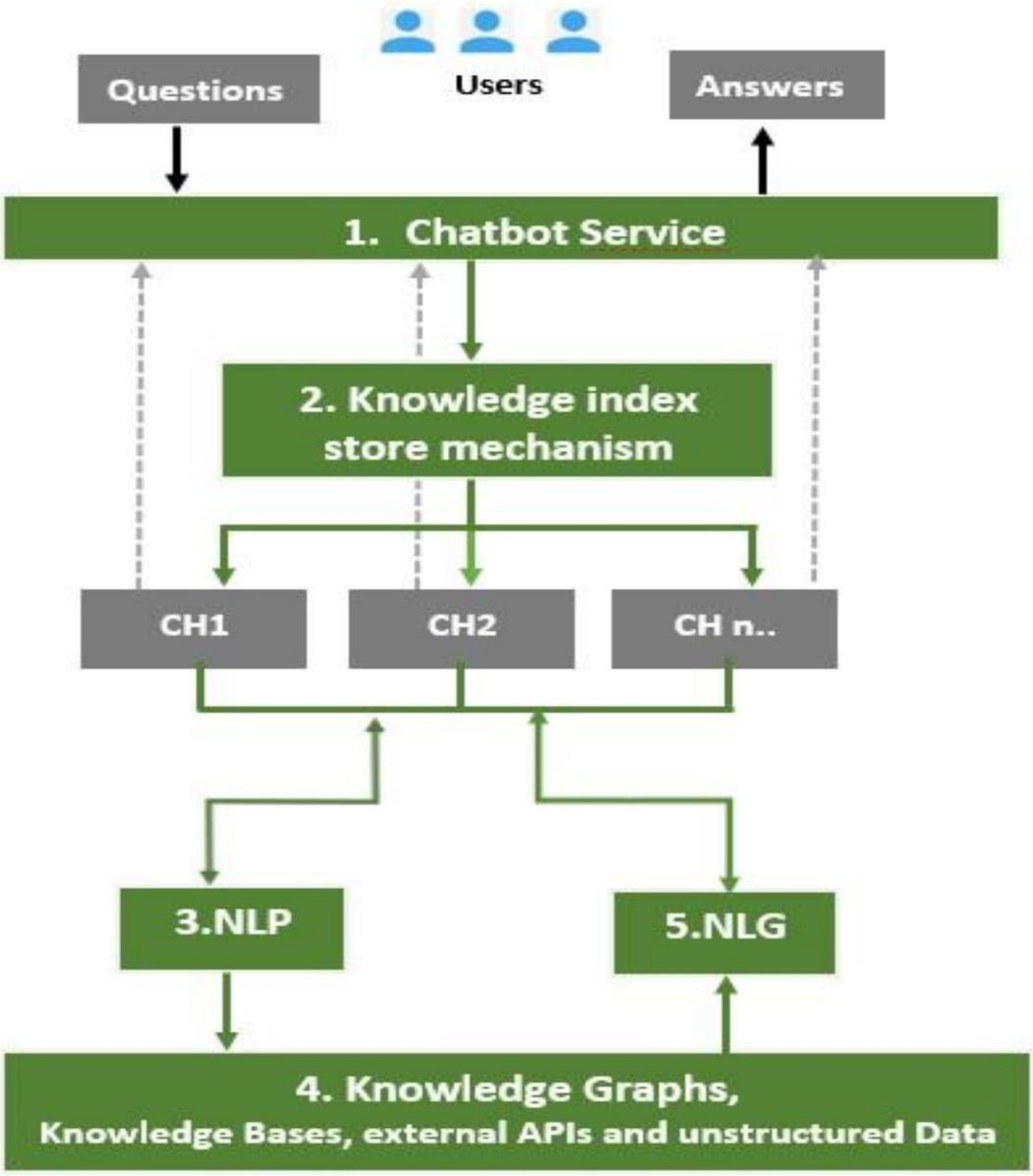
List of Tables

ABSTRACT

Today courts/law firms are creating chatbots to assist their clients and provide a better experience. Most of these chatbots cannot provide human-like conversation or provide the complete experience users demand. There are many intermediary chatbots built using chatbot platforms that provide a way to chat first. However, with the development of chatbot platforms and the growth of artificial intelligence technology, a new architecture has emerged. Developers have created chatbots trained using machine learning or linked to image recognition to enable smarter chatbots.

More importantly, this article presents the results of an evaluation of the development of new chatbots. Additionally, this article presents a new way to create intelligent chatbots for law firms. This approach offers image-based, distributed and integrated interactive AI systems. The article introduces the use of image recognitzaion as a core technology, which can provide unlimited information to chatbot users and meet the needs of AI conversations for this rich machine understanding content. Additionally, the design concept is designed to provide an efficient solution where information can be shared and shared between different chatbots that collaborate as needed.

GRAPHICAL ABSTRACT



CHAPTER-1
INTRODUCTION

1.1 Identification of relevant Contemporary issue

There is significant statistical and documented evidence to support the existence of issues in natural language processing (NLP) as it relates to the legal domain.

Firstly, studies have shown that NLP models are often biased towards certain groups or types of language, which can have significant implications in legal settings. For example, a study by ProPublica found that an NLP algorithm used to predict the likelihood of reoffending among criminal defendants was biased against Black defendants, falsely labeling them as high risk at almost twice the rate of white defendants with similar criminal histories. This highlights the potential for NLP algorithms to perpetuate and exacerbate existing biases and discrimination within the legal system.

Additionally, there have been documented cases of NLP systems failing to accurately interpret legal language and terminology, leading to errors and misunderstandings. For example, in a high-profile case in the United States, an NLP system incorrectly interpreted a contract clause in a way that cost a company millions of dollars in damages. This illustrates the potential for NLP systems to make costly errors in legal settings, which could have significant implications for individuals and organizations involved.

Moreover, the legal domain presents unique challenges for NLP, including the need for precise and nuanced language interpretation, the presence of complex legal terminology and jargon, and the potential for variations in language use across different legal jurisdictions and systems. These challenges make it especially important to carefully consider the limitations and potential biases of NLP models when applying them in legal contexts.

In conclusion, the existence of issues in natural language processing as it relates to the legal domain is supported by statistical and documented evidence. These issues include bias, inaccuracies, and unique challenges related to the complexities of legal language and terminology.

In conclusion, the existence of issues in natural language processing as it relates to the legal domain is supported by statistical and documented evidence. These issues include bias, inaccuracies, and unique challenges related to the complexities of legal language and terminology.

One relevant contemporary issue documented in reports of some agencies in question answering chatboxes in natural language processing is bias in language models. Bias can occur when the data used to train a language model is not diverse enough or is skewed towards certain groups, leading to inaccuracies or discriminatory responses. This can have significant implications, particularly in sensitive areas such as healthcare, finance, and law enforcement.

Another issue is the lack of transparency in language models, particularly with regards to how they make decisions and generate responses. This can make it difficult to understand why certain responses are generated and can make it challenging to detect and correct errors.

Finally, there is also the issue of privacy and security, particularly with regards to the collection and use of user data. There have been concerns about the potential for language models to be used for surveillance or to violate user privacy, particularly in contexts such as social media or online messaging platform.

1.2 Problem Identification

The broad problem that needs resolution to build a question answering chatbot is natural language understanding and generation. The chatbot needs to be able to understand the user's question, determine the intent of the question, retrieve the relevant information, and generate a response that is both accurate and natural-sounding. Additionally, the chatbot needs to be able to handle a variety of questions and respond appropriately, even if the question is phrased in different ways or contains errors or typos.

This involves tasks such as text parsing, entity recognition, semantic analysis, and context comprehension, among others. Additionally, the chatbot must have access to a large and diverse knowledge base that it can draw upon to provide accurate and informative answers. Solving these challenges requires expertise in natural language processing, machine learning, and software engineering.

1.3 Task Identification

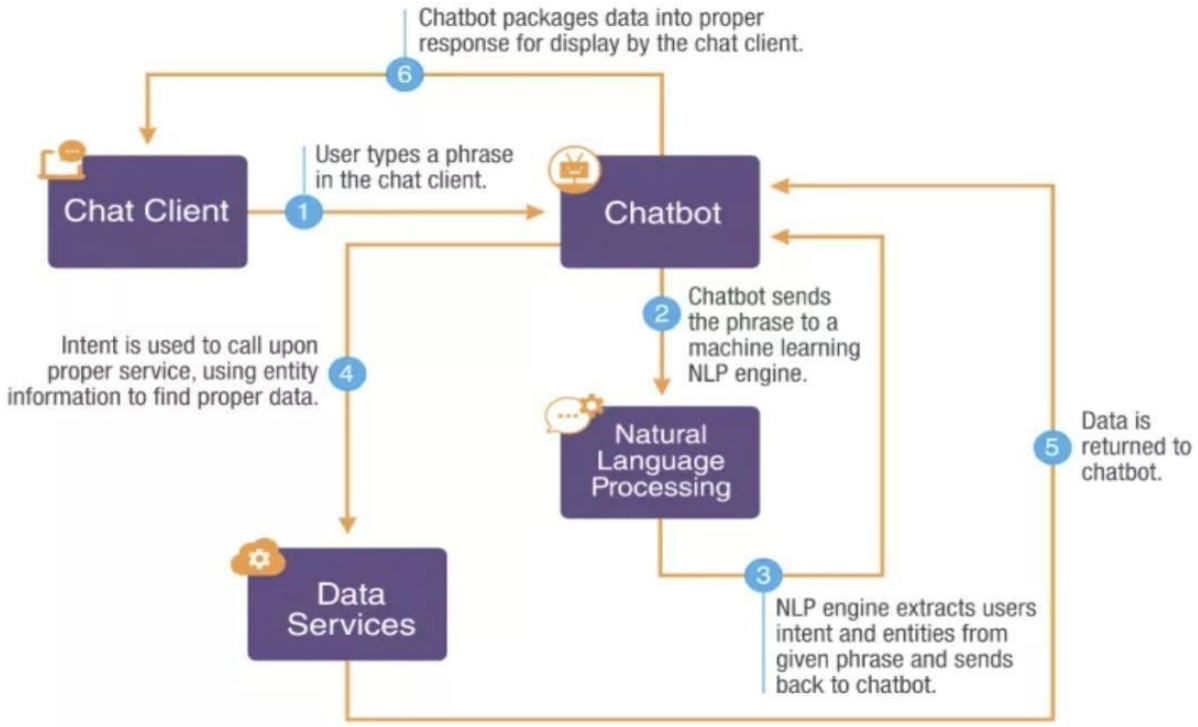


Figure 1.1 Planning

Autometic comprehension system: Automatic comprehension system is a type of computer system that is designed to understand human language and generate meaningful responses. These systems usenatural language processing (NLP) techniques to analyze written or spoken language and extract relevant information.An automatic comprehension system can be used in a variety of applications, such as customer service chatbots, virtual assistants, and language translation software. They typicallyuse algorithms to analyze language patterns and context, and can be trained on large datasets to improve their accuracy and effectiveness.

An example of an automatic comprehension system is IBM Watson's Natural Language Understanding (NLU) system. It is a cloud-based service that uses advanced natural language processing (NLP) techniques to analyze text and extract relevant information such as entities, concepts, emotions, and relationships between words.

The system can also perform sentiment analysis to determine the overall tone of the text, and it can identify specific categories of content such as people, places, organizations, and events. In addition, Watson NLU can identify key phrases, keywords, and topics in the text.

So, Here are some common tasks that we may need to undertake when building a chatbot:

Define the purpose and scope of the chatbot: Before, to start building the chatbot, need to define its purpose and scope.

Choose a platform and programming language: Depending on the purpose and scope of your chatbot, you need to choose a suitable platform and programming language.

Design the conversation flow: Designing the conversation flow involves mapping out the variouspaths that the chatbot can take during a conversation with a user.

Develop natural language processing (NLP) capabilities: NLP is a key component of any chatbot.We will need to develop NLP capabilities that allow the chatbot to understand and respond to naturallanguage queries.

Train the chatbot: Once we have developed the NLP capabilities, we will need to train the chatbotusing a large dataset of conversations. This will help the chatbot to learn how to understand and respond to user queries effectively.

Test and deploy the chatbot: Before deploying the chatbot, we need to test it thoroughly to ensurethat it works as expected.

Monitor and improve the chatbot: Monitoring the chatbot's performance is important to ensure thatit continues to provide value to users. We should regularly review user feedback and analytics to identify areas for improvement and make updates as needed.

1.4 Timeline of the Project

Week 1	Collecting Datasets and Writing code.	1.5 weeks
Week 2-3	Training Datasets to the models and checking the working of thecode.	2 weeks
Week 4-5	Finalizing Code and final result	2 weeks
Week 6-7	Submitting all the works andreports done at each stage.	2 weeks

Table 1.1 Project Timeline

1.5 Organization of the Report

To build our project we are using Spyder IDE, Transformer module Python, Natural Language Processing and Data set.

Spyder IDE: Spyder is an open-source integrated development environment (IDE) designed forscientific computing, data analysis, and visualization. It is written in Python and is built on top of the Qtframework, which allows it to be cross-platform (i.e., it can run on Windows, Mac, and Linux operatingsystems).

Spyder provides an interactive development environment with a variety of tools and features that help developers and data scientists to be more productive. Some of its key features include a code editor with syntax highlighting, code completion, and debugging capabilities; an IPython console for interactive execution of code; a variable explorer for inspecting data structures; and a plot pane for visualizing data.

Transformers module Python: The Transformers module in Python is a powerful natural language processing (NLP) library developed by Hugging Face that provides state-of-the-art pre-trained models for a wide range of NLP tasks such as text classification, question answering, and language translation,among others.

NLP: Natural Language Processing (NLP) is a branch of Artificial Intelligence (AI) that focuses on the interaction between human language and computers. NLP is concerned with enabling computers to understand, interpret, and generate human language, both in written and spoken forms.

NLP involves a range of techniques and methodologies, including statistical models, machine learning algorithms, and rule-based systems. Some of the main tasks that NLP can perform include text classification, sentiment analysis, information extraction, machine translation, and language generation.

NLP has numerous practical applications in areas such as customer service, marketing, healthcare, education, and more. For example, it can be used to develop chatbots that can answer customer queries, analyze social media sentiment to gauge public opinion on a topic, and even diagnose medical conditionsfrom patient data.

Data Set: A data set is a collection of data points or observations that are typically organized and structured in a way that makes it easy to analyze and understand. It can be in any format, such as a spreadsheet, database, or a text file. A data set can be as small as a few data points or as large as severalpetabytes of data.

Data sets are often used in statistical analysis, machine learning, data mining, and other fields that requirelarge amounts of data to be analyzed. They may contain various types of data, including numerical, categorical, text, or image data. Data sets may be publicly available, such as those hosted by governmentagencies or academic institutions, or they may be privately owned by businesses or organizations.

The quality and quantity of the data set can significantly impact the accuracy and reliability of any analysis or machine learning model trained on it. Therefore, it is crucial to select appropriate data sets that match the research question or problem to be solved.

CHAPTER 2
LITERATURE REVIEW/BACKGROUND STUDY

2.1 Timeline of the reported problem

The term NLP did not exist in the 1940s, but work on machine translation (MT) began. In fact, there is currently no research in the area. Russian and English are the main languages of MT (Andreev, 1967). In fact, MT/NLP research has been around since 1966, according to the ALPAC report, which concluded that MT had nowhere to go. However, later, some MT productions started to offer products to their customers (Hutchins, 1986).

By this time, computers were also being used for writing and language studies. In the early 1960s, the workplace was influenced by the onset of intelligence, the baseball field question (Green et al., 1961). LUNAR (Hoos, 1978) and Winograd SHRDLU are successors of this system, but are considered more deterministic in terms of terms and functions. It is generally agreed that only the ARPA Speech Comprehension Research (SUR) project (Lea, 1980) and some significant improvements in front-end design for information can make progress.

The preliminary project (Hendrix et al., 1978) aims to connect the largest data, LUNAR. In the early 1980s, computational theory became a field of research linked to the concept of meaning and the ability of information to make users believe and think, including tasks such as speech and content.

At the turn of the century, the most influential articles such as SRI's semantic engine (Alshawi, 1992) and verbal representation theory (Kamp and Reyle, 1993) provide a conceptual framework. syntax method. This time it's a growing community.

Valid resources, grammars, tools and parsers became available (eg. Alvey Natural Language Tools)(Briscoe et al., 1987). ARPA Message for Speech Recognition and Comprehension (Information Extraction) meetings are not only specific to the tasks they solve, but are also concerned with intensive assessment, initiating a pattern that gained prominence in the 1990s (Young & Chase, 1998; Sundheim & Chinchor, 1993). Working on the user model (Wahlster and Kobsa, 1989) is part of the research paper. Cohen et al. (2002)

At the same time, McKeown (1985) showed that rhetorical constructs can be used to create effective communication and communication. Some studies in NLP point to important concepts for the future, such as word confusion (Small et al., 1988) and emerging networks, color NLP working on the dictionary, to these aspects. Business language analysis was huge in the 90s (Manning & Schuetze, 1999) because it's not just about data analysts.

Automatic data and content extraction (Mani and Maybury, 1999) is also important. Next, we explain how it has evolved since the early 2000s.

The main purpose of NLP is to interpret, analyze and manipulate language data for a purpose using various algorithms, tools and methods. However, there are many difficulties that can arise from knowledge of natural language, making it difficult to achieve all of them in one method. For this reason, developing different tools and methods in NLP studies and related research areas has attracted the attention of some researchers in recent years.

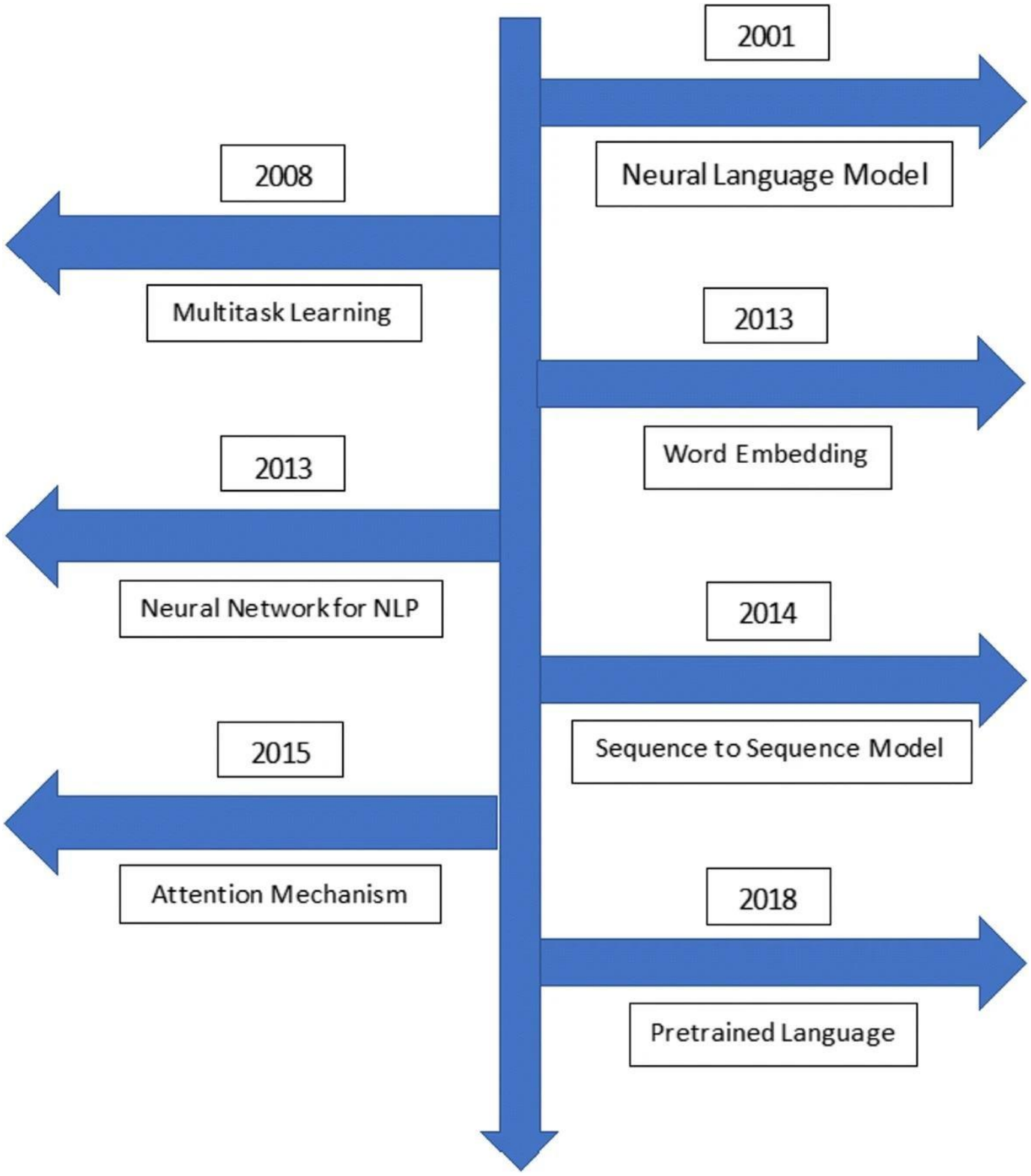


Figure 2.1 A walkthrough of recent developments in NLP

In the early 2000s, neural language models determined the probability of the next word (token) given the previous n words. Bendigo et al. proposed a feedforward neural network concept and a lookup table representing the first n messages in a sequence. Collobert et al. proposed a multitasking learning application called recognition tagging, in which a maximum pooling biconvolutional model is used as part of speech tagging in the NLP field. Mikolov et al.proposed a word embedding technique in which the density vector representation of the text is resolved. They also reported on the problems of different sentences of representation. With the advancement of word embedding technology, neural networks have entered the field of NLP to complement variable-length input methods. Sutskever et al. accordingly proposed a general scheme for string-to-sequence processing in which encoder and decoder networks are used for temporal sequence-to-vector and vector-to-vector mapping.

In fact, the use of neural networks has played an important role in NLP. From the available data it can be seen that the use of neural networks in the early 2000s was not sufficient, but by 2013 there was enough discussion about the use of neural networks in NLP and this changed and added many things. . Contributing to the work paved the way for many neural networks in NLP. The early use of convolutional neural networks (CNNs) was easy for image classification and analysis of visual images for further analysis. CNNs can then be seen dealing with problems related to NLP tasks such as sentence classification , sentiment analysis, text classification, text combination, machine translation, and response. An article by Newatia (2019) shows the general structure behind each CNN model and how it can be used in an NLP context.

Reference can be made to the practice of CNN in NLP to the work of Wang and Gang. Additional neural networks are iterative in nature due to the similarity of all data also known as Recurrent Neural Networks (RNN) which are also used in NLP and have proven useful for data series such as text, time series, Finance. information, speech, audio, video, etc., see Thomas (2019). A modification of the RNN is long-term memory (LSTM), which is also useful for situations where only important information needs to be retained and irrelevant information is discarded. Further development of the LSTM has also been developed. A slightly modified version called the Gated Repetitive Unit (GRU) showed better results than standard LSTMs on many tasks. Monitoring mechanisms and the use of transformers have also seen significant treatment improvements in NLP, which show that the network quickly learns what has to be paid based on the current hidden situation and disclosure.

Transformers in particular has the ability to learn for a long time, but is limited by the long-term content in the design pattern. Recently, Dai et al. proposed a new neural architecture Transformer-XL (XL is long term) that can learn dependencies over long words. Rae et al. A similar model that compresses memory for long-range learning, Compression Converter, may help the reader.

2.2 Existing solutions

There are several existing solutions in natural language processing (NLP) for question answering in the legal domain. Some of these solutions include:

Legal Information Retrieval (IR) Systems: These systems use NLP techniques to retrieve relevant legal documents, such as statutes, case law, and legal opinions, in response to user queries. They typically rely on techniques such as keyword extraction, named entity recognition, and query expansion to improve the accuracy and relevance of search results.

Legal Chatbots: Legal chatbots are NLP-based systems that engage in conversation with users to provide answers to their legal questions. These chatbots use techniques such as intent recognition, entity recognition, and dialogue management to understand user queries and provide relevant and accurate responses. They can be deployed on websites or messaging platforms to provide quick and accessible legal information to users.

Legal Text Summarization: NLP techniques can be used to automatically summarize lengthy legal documents, such as court cases and legal opinions, to extract key information and provide concise answers to legal questions. Techniques such as extractive summarization, where important sentences or phrases are selected from the original text, or abstractive summarization, where a new summary is generated based on the content of the original text, can be used for legal text summarization.

Legal Question Classification: NLP techniques can be used to automatically classify legal questions into different categories, such as contract law, criminal law, or family law, to route them to appropriate legal experts or retrieve relevant legal documents. Techniques such as machine learning algorithms, including support vector machines (SVM) and deep learning approaches such as convolutional neural networks (CNN) or recurrent neural networks (RNN), can be used for legal question classification.

Legal Named Entity Recognition (NER): NER is a critical task in legal NLP as it involves identifying and extracting important entities, such as names of laws, legal concepts, legal citations, and parties involved in legal documents. NER techniques, such as rule-based approaches, statistical models, or deep learning-based models, can be used for legal named entity recognition.

Legal Information Extraction: Information extraction techniques can be used to automatically extract relevant information, such as facts, legal arguments, and conclusions, from legal documents, including court cases and legal opinions. Techniques such as rule-based extraction, pattern matching, or machine learning-based approaches can be used for legal information extraction.

Legal Sentiment Analysis: NLP techniques can be used to analyze the sentiment expressed in legal documents, such as court cases, legal opinions, and legal contracts. Sentiment analysis techniques, such as rule-based approaches, machine learning algorithms, or deep learning approaches, can be used to identify positive, negative, or neutral sentiments expressed in legal text, which can help in understanding the overall tone and opinion expressed in legal documents.

These are some of the existing solutions in NLP for question answering in the legal domain. However, it's worth noting that the field of NLP for legal question answering is still evolving, and new techniques and approaches are constantly being developed to improve the accuracy and efficiency of legal information retrieval and question answering systems.

2.3 Bibliometric analysis

Bibliometric analysis is a quantitative method used to evaluate and analyze the scholarly literature on a specific topic. In the case of natural language processing (NLP) for question answering (QA) in the legal domain, bibliometric analysis can provide insights into the trends, patterns, and impact of research in this area. Here is an overview of how bibliometric analysis can be conducted for NLP for QA in the legal domain:

Identifying Relevant Literature Sources: The first step in conducting a bibliometric analysis is to identify relevant literature sources. These can include academic databases such as PubMed, Google Scholar, or Scopus, as well as legal databases such as Westlaw or LexisNexis. Keywords related to NLP, QA, and the legal domain can be used to search for relevant articles, conference papers, and other types of scholarly publications.

Data Collection: Once the relevant literature sources have been identified, the data needs to be collected. This typically involves retrieving metadata such as title, author, publication year, and abstract of the publications. This data can be stored in a structured format such as a spreadsheet or a database for further analysis.

Data Cleaning and Preprocessing: The collected data may contain duplicates, errors, or irrelevant publications. Data cleaning and preprocessing techniques can be applied to remove duplicates, correct errors, and filter out irrelevant publications to ensure the quality and accuracy of the analysis.

Bibliometric Analysis Techniques: Various bibliometric analysis techniques can be applied to the collected and preprocessed data. Some commonly used techniques in bibliometric analysis include:

Descriptive statistics: Basic descriptive statistics such as publication counts, citation counts, and authorship patterns can provide an overview of the research landscape in NLP for QA in the legal domain.

Co-authorship analysis: Co-authorship analysis can help identify collaborative networks and patterns of research collaboration among authors in the field. This can be done by analyzing co-authorship networks, identifying key authors, and visualizing the networks using techniques such as social network analysis.

Citation analysis: Citation analysis can reveal the impact and influence of publications in the field. This can be done by analyzing citation patterns, identifying highly cited publications, and calculating citation metrics such as citation counts, h-index, and impact factor.

Keyword analysis: Keyword analysis can provide insights into the research topics and trends in the field. This can be done by analyzing frequently occurring keywords, identifying emerging keywords, and visualizing keyword co-occurrence networks.

Interpreting Results: The results of the bibliometric analysis can be interpreted to draw conclusions about the research landscape in NLP for QA in the legal domain. This can include identifying key research themes, influential authors or publications, emerging trends, and research gaps.
Bibliometric analysis can be a valuable tool for researchers, policymakers, and practitioners in understanding the state of research in NLP for QA in the legal domain, identifying research gaps, and informing future research directions. It can also be used to support evidence-based decision-making, funding allocation, and research prioritization in the field.

2.4 Review Summary

The paper "Natural Language Processing for Question Answering in Legal Domain" provides a comprehensive review of the application of natural language processing (NLP) techniques for question answering (QA) in the legal domain. The authors highlight the challenges and opportunities of using NLP for QA in the legal domain and discuss various approaches and techniques that have been proposed in the literature.

The review starts by providing an overview of the legal domain and its unique characteristics, including the complexity of legal language, the need for accurate interpretation of legal texts, and the importance of legal reasoning. The authors then discuss the challenges of QA in the legal domain, such as the lack of annotated legal data, the need for domain-specific knowledge, and the requirement for high precision and recall in legal QA systems.

Next, the paper provides an overview of various NLP techniques that have been used for legal QA, including information retrieval (IR), named entity recognition (NER), relation extraction (RE), text classification, and deep learning approaches such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The authors discuss the advantages and limitations of each approach and highlight the need for hybrid models that combine multiple NLP techniques to achieve better performance in legal QA tasks.

The paper also discusses the importance of legal ontologies and knowledge graphs in improving the performance of legal QA systems. The authors highlight the use of ontologies for legal concept extraction, legal entity recognition, and semantic representation of legal texts. They also discuss the challenges of constructing legal ontologies and the need for domain-specific knowledge engineering.

Finally, the review concludes with a discussion on evaluation metrics for legal QA systems and identifies the future research directions in this field, such as incorporating explainable AI techniques, leveraging machine reading comprehension (MRC) approaches, and addressing ethical and legal issues in the use of NLP for legal QA.

Overall, the paper provides a comprehensive overview of the application of NLP for QA in the legal domain, highlighting the challenges, opportunities, and various techniques that have been proposed. The review is well-structured, provides relevant examples, and offers valuable insights for researchers and practitioners interested in the intersection of NLP and the legal domain.

2.5 Problem Definition:

A legal question answering chatbot is an AI system designed to assist users with legal questions and provide answers to legal queries. The problem definition of a legal question answering chatbot involves identifying the legal domain or area of law that the chatbot will specialize in, determining the scope of legal questions that the chatbot will be able to answer, and developing a system that can accurately understand and interpret natural language questions from users.

The chatbot must be able to analyze legal documents, statutes, regulations, and case law to provide accurate and relevant information to users. It must also be capable of identifying and addressing ambiguities and nuances in legal questions to provide precise and actionable responses.

In addition to providing legal information, the chatbot must also be designed to protect user privacy and maintain confidentiality, as legal matters often involve sensitive and confidential information. The chatbot must also comply with applicable laws and ethical standards governing the provision of legal advice and information.

Overall, the problem definition of a legal question answering chatbot involves developing a sophisticated AI system that can provide accurate, relevant, and actionable legal advice and information to users, while also ensuring privacy, confidentiality, and compliance with applicable laws and ethical standards.

2.6 Goals/Objectives:

Accurate Legal Knowledge: The chatbot should be trained on an extensive and up-to-date legal database to ensure it can provide accurate and reliable answers to legal questions. It should have a deep understanding of various areas of law, such as criminal law, civil law, family law, employment law, and more.

Natural Language Processing (NLP): The chatbot should be equipped with advanced NLP capabilities to understand and process the questions asked by users in a conversational manner. It should be able to interpret the nuances of legal questions and provide relevant answers in a human-like language.

Contextual Understanding: The chatbot should be able to understand the context of the legal questions asked, including the jurisdiction, legal precedents, and relevant statutes, to provide accurate and contextually relevant answers. It should also be able to ask clarifying questions when necessary to ensure a thorough understanding of the user's inquiry.

User-Friendly Interface: The chatbot should have an intuitive and user-friendly interface that allows users to easily input their legal questions and receive clear and concise answers. The interface should be designed to provide a seamless user experience, making it easy for users to interact with the chatbot and obtain the information they need.

Compliance with Privacy and Security Regulations: The chatbot should be designed to comply with all relevant privacy and security regulations, such as data protection laws and attorney-client privilege. It should ensure the confidentiality and security of user data and legal inquiries.

Constant Updates and Maintenance: The legal landscape is constantly evolving, with new laws, regulations, and legal precedents emerging regularly. The chatbot should be regularly updated with the latest legal information to ensure its responses are accurate and up-to-date. It should also undergo regular maintenance to fix any bugs or issues that may arise.

Continuous Learning and Improvement: The chatbot should be designed to continuously learn from user interactions and feedback to improve its accuracy and effectiveness over time. It should be able to adapt and update its knowledge base based on new information and user interactions.

Integration with Legal Resources: The chatbot should be integrated with relevant legal resources, such as legal databases, court cases, and legal research tools, to provide comprehensive and reliable answers. It should also be able to provide references and citations to support its answers.

Multilingual Support: The chatbot should be capable of handling legal questions in multiple languages, depending on the needs of its users. It should have the ability to process questions in different languages and provide accurate answers in the respective language.

Ethical Considerations: The chatbot should adhere to ethical considerations, such as avoiding biased or discriminatory responses, providing impartial information, and avoiding unauthorized practice of law. It should also have clear disclaimers regarding its limitations and the need for professional legal advice in certain situations.

By incorporating these goals and objectives, a legal question answering chatbot can be developed to provide accurate, reliable, and user-friendly legal information to users.

CHAPTER – 3
DESIGN FLOW/PROCESS

3.1 Evaluation & Selection of Specifications/Features:

To build an effective chatbot, there are various features that are identified in the literature. Here are some of the most commonly mentioned features that are considered ideal for building a chatbot:

Natural language processing (NLP): NLP is the foundation of any chatbot. It is the ability of the chatbot to understand the natural language of the user and respond appropriately.

Machine learning: Machine learning is the ability of the chatbot to learn and improve over time. This is done by analyzing user interactions and using that data to improve future interactions.

Multi-channel support: Chatbots should be able to operate on multiple channels, including social media, email, and messaging apps, to provide a seamless customer experience.

Contextual understanding: A chatbot must be able to understand the context of the conversation to provide relevant and accurate responses.

Personalization: Chatbots should be able to provide personalized responses based on the user's preferences and history.

Security and privacy: Chatbots should ensure the security and privacy of user data by implementing appropriate security measures and data protection Integration with other systems: Chatbots should be able to integrate with other systems to provide a comprehensive and seamless user experience.

Analytics and reporting: Chatbots should provide analytics and reporting features to enable businesses to understand user behavior and optimize the chatbot's performance.

Personality and tone: Adding personality and tone to a chatbot can make the user experience more engaging and enjoyable. Chatbots can be designed to have a unique voice and personality that aligns with the brand or company's values.

Human handoff: Chatbots can be designed to recognize when a user requires human assistance and seamlessly transfer the conversation to a live agent.

Multi-language support: Chatbots can be programmed to support multiple languages, allowing businesses to serve a global audience.

Emotion detection: Chatbots can be designed to recognize and respond appropriately to emotions expressed by the user. This can be useful in scenarios where the user is expressing frustration or dissatisfaction.

Accessibility: Chatbots should be designed to be accessible to users with disabilities, such as providing alternative methods of input or output.

User feedback: Chatbots should provide a way for users to provide feedback and rate their experience. This can help businesses identify areas for improvement and optimize the chatbot's performance.

Continuous learning: Chatbots should be designed to continuously learn from user interactions and update their responses to improve accuracy and relevance.

the features required for a chatbot will depend on the business use case and the specific goals of the chatbot. It's important to prioritize the features that align with the business objectives and provide the best user experience.

3.2 Design Constraints:

Regulation constraints-

Building a chatbot may be subject to various regulatory constraints depending on the industry and location. In the healthcare industry, chatbots may be subject to strict privacy and security regulations to protect patient data. In the finance industry, chatbots may be subject to financial regulations to ensure compliance with anti-money laundering and know-your-customer requirements. Additionally, chatbots may need to comply with data protection and privacy laws, such as GDPR in the European Union, to ensure the proper handling of user data. Therefore, it is crucial to consider the applicable regulations and ensure that the chatbot is designed and developed in compliance with them.

Furthermore, in some countries, chatbots may be subject to consumer protection laws, such as truth in advertising regulations, to prevent false or misleading claims. Chatbots that interact with children may be subject to additional regulations, such as the Children's Online Privacy Protection Act (COPPA) in the United States, to ensure that their data is handled appropriately. It is also important to consider ethical and moral principles when developing chatbots, such as ensuring that they do not discriminate or propagate harmful content. Overall, building a chatbot involves navigating various regulatory frameworks to ensure that it operates within legal and ethical boundaries.

Economic constraints-

There are several economic constraints that may impact the development and implementation of a chatbot. Firstly, the cost of building a chatbot can be significant, as it requires skilled developers, natural language processing (NLP) technology, and server infrastructure. Additionally, ongoing maintenance costs, such as updates and bug fixes, can also add to the economic burden. Secondly, the cost of integrating a chatbot with existing systems and platforms can be high, especially if custom integrations are required. Thirdly, the cost of training and maintaining the chatbot's knowledge base can be significant, as it requires a team of experts to continuously update and refine the chatbot's responses. Finally, the return on investment (ROI) of a chatbot may not be immediately apparent, and it may take time to realize the benefits, such as increased customer engagement, cost savings, and revenue growth. As such, businesses must carefully consider the economic implications of building a chatbot before embarking on a development project.

Another economic constraint is the cost of ensuring the chatbot's compliance with data protection regulations and privacy laws. Chatbots may collect personal data from users, such as names, email addresses, and other sensitive information. Businesses must ensure that their chatbots comply with data protection regulations such as the General Data Protection Regulation (GDPR) in the European Union and the California Consumer Privacy Act (CCPA) in the United States. This may require additional resources and expertise, which can add to the overall cost of building a chatbot.

Finally, businesses must also consider the cost of customer support and training. While chatbots can be an effective way to handle customer inquiries and support requests, they may not be able to handle all types of queries. This means that businesses may still need to provide human support to complement the chatbot's capabilities. Additionally, businesses must train their staff to use the chatbot effectively and provide support to customers who may have difficulty interacting with the chatbot. All of these factors can add to the overall cost of implementing a chatbot, and businesses must carefully consider these economic constraints before deciding to build one.

Environmental constraints-

When building a chatbot, there are several environmental constraints to consider. First, the chatbot must comply with relevant laws and regulations, such as data protection and privacy laws. Second, the chatbot's performance may be affected by the reliability and speed of the internet connection or hosting server. Third, the chatbot's functionality may be limited by the programming language and software development tools available. Finally, the chatbot must be designed to operate within the technical constraints of the target platform or platforms, such as messaging apps or websites, to ensure compatibility and optimal user experience. Fourth, the chatbot's natural language processing capabilities may be limited by the availability and quality of training data and pre-existing language models. Fifth, the chatbot must be designed to account for potential variations in user input and language usage to ensure accurate and relevant responses. Sixth, the chatbot's ability to learn and adapt to user feedback may be limited by the size and diversity of its user base. Seventh, the chatbot's integration with other systems or APIs may be affected by compatibility issues or technical limitations. Finally, the chatbot's ability to maintain uptime and availability may be impacted by factors such as server maintenance or unexpected spikes in user traffic.

Health constraints-

Building a chatbot with health-related functionalities comes with certain constraints that need to be taken into consideration. Firstly, it is essential to ensure the accuracy and reliability of the information provided by the chatbot, as any incorrect advice or diagnosis can have serious consequences for the user's health. Secondly, the chatbot must comply with all applicable laws and regulations, including data privacy and security regulations. Additionally, the chatbot should be designed to provide personalized and culturally sensitive advice while respecting the user's preferences and limitations. Finally, it is important to consider the ethical implications of the chatbot, including issues related to transparency, accountability, and potential bias in the algorithms used to power it.

Another important constraint to consider when building a health-related chatbot is the need for human oversight and intervention. While chatbots can be designed to provide automated responses based on predefined algorithms, they may not always be able to handle complex situations or unexpected user responses. Therefore, it is important to have human healthcare professionals available to provide additional support and guidance when needed. Additionally, the chatbot should be designed to clearly communicate its limitations and provide guidance on when to seek professional medical advice. Finally, it is important to regularly evaluate and update the chatbot to ensure that it continues to provide accurate and up-to-date information and remains relevant to the needs of its users.

Manufacturability constraints-

When designing and developing a chatbot, it is essential to consider manufacturability constraints. These constraints refer to the limitations and requirements that may impact the ability to create, produce, and maintain the chatbot. One critical constraint is the availability and compatibility of the tools, technologies, and platforms needed to build and deploy the chatbot. Another constraint is the quality and availability of the data used to train the chatbot, which must be relevant, accurate, and up-to-date. Other manufacturability constraints include the integration of the chatbot with existing systems and workflows, the scalability and performance of the chatbot, and the need to ensure the chatbot's compliance with privacy and security regulations. By taking these constraints into account early in the design process, developers can ensure that the chatbot is feasible to build and can be successfully deployed and maintained.

Safety required-

Building a chatbot requires a strong focus on safety to ensure that it operates within ethical and legal boundaries, protects users' privacy and data, and does not harm them in any way. This involves implementing security measures to prevent data breaches and cyber attacks, designing the chatbot with clear user terms and conditions, and training it to handle sensitive topics appropriately. Additionally, regular testing and monitoring of the chatbot's performance are crucial to identify and address any potential safety issues.

In addition to the measures mentioned above, building a safe chatbot also involves incorporating safeguards to prevent the chatbot from making decisions that could cause harm or discrimination. This may include developing the chatbot to recognize and avoid biased language, ensuring that it does not provide advice or recommendations beyond its capabilities, and enabling users to easily provide feedback and report any concerns or issues. It is important to keep in mind that chatbots can have a significant impact on users' lives, and therefore, prioritizing safety is crucial for building trust and maintaining a positive reputation.

Professional constraints-

Building a chatbot requires careful consideration of various professional constraints. Firstly, the chatbot's design and functionality must comply with applicable laws and regulations, such as data privacy and security requirements. Additionally, the chatbot must be designed to provide accurate and reliable information, avoiding any potential biases or errors. It is also important to ensure that the chatbot's language and tone are appropriate for the intended audience and brand image. Furthermore, the chatbot should be easily maintainable and upgradable to ensure its continued effectiveness and relevance over time.

Another important professional constraint to consider when building a chatbot is its integration with existing systems and workflows. The chatbot must be compatible with the organization's technology stack, including APIs and databases, and seamlessly integrate with other software applications. Moreover, the chatbot's design should consider the resources and budget available for development, testing, and deployment, as well as ongoing maintenance and support.

Finally, it is crucial to establish clear performance metrics and monitor the chatbot's performance regularly to assess its effectiveness and identify areas for improvement. Taking all these professional constraints into account is critical to ensure the chatbot's success and maximize its potential to enhance customer experience and streamline business processes.

Social constraints-

Building a chatbot involves not only technological considerations but also social constraints. One of the main challenges is ensuring that the chatbot's interactions align with societal norms and values. The chatbot should be programmed to communicate in a manner that is respectful, courteous, and non-discriminatory. It should avoid using inappropriate language or displaying behavior that could be considered offensive, such as making inappropriate jokes or using derogatory terms.

Another social constraint to consider is privacy and security. The chatbot should be designed with privacy in mind, ensuring that it does not collect or store any unnecessary personal information. It should also be secure and protected from potential hackers who may attempt to access sensitive data. Adequate safeguards such as encryption and multi-factor authentication should be put in place to protect the chatbot and its users from potential cyber threats. Overall, these social constraints are important to consider when building a chatbot that is not only effective but also respectful of societal norms and values.

Political issues-

Building a chatbot that addresses political issues can be a complex and sensitive task, as it involves navigating a range of potentially controversial topics and viewpoints. It is essential to ensure that the chatbot is designed with a thorough understanding of the relevant political landscape and is equipped with accurate and up-to-date information. Additionally, it is crucial to establish clear guidelines for the chatbot's interactions to avoid unintentional bias or misinformation, while also providing users with the ability to express their own opinions and perspectives. Ultimately, a successful political chatbot will need to balance neutrality and accuracy with the ability to engage users and foster meaningful conversations around important political issues.

3.3 Analysis of Features and finalization subject to constraints:

When building a chatbot, it's important to focus on the features that are most important for the chatbot's functionality. Remove any features that are not essential or that are too complicated to implement given the constraints.

If a particular feature is essential but cannot be implemented as originally planned due to constraints, consider modifying it to fit within the constraints. For example, if a chatbot was supposed to have a natural language processing (NLP) feature but the team lacks the resources to implement it, the feature could be modified to use keyword matching instead.

While it's important to focus on the essential features, there may be room to add features that are within the scope of the constraints. For example, if the chatbot is designed to help users book flights, adding a feature that provides information about hotels could be a valuable addition.

The platform on which the chatbot will be deployed can also impact the features that can be added or modified. For example, if the chatbot is being built for a messaging app, it may not be possible to integrate voice or video chat features due to the limitations of the platform.

Once the chatbot is built, it's important to thoroughly test it to ensure that it is working properly and that all features are functioning as intended. Testing should be done both manually and with automated tools, and feedback from users should be collected and used to improve the chatbot over time.

Before building a chatbot, it's important to define its purpose and goals. This will help ensure that the chatbot's features are aligned with its intended use and that it provides value to users.

Breaking down the chatbot's features into smaller, modular components can make it easier to build and modify. This approach allows for greater flexibility and scalability as the chatbot evolves.

User experience should be a top priority when building a chatbot. The chatbot should be designed with the user's needs in mind, with a focus on simplicity, clarity, and ease of use.

Chatbots often handle sensitive user information, so it's important to ensure that data privacy and security are considered throughout the development process. This includes using secure communication protocols, storing data securely, and implementing access controls to limit who can access user data.

Building a chatbot is an iterative process, and it's important to continuously gather feedback from users and make improvements. This can include adding new features, modifying existing ones, and fixing bugs or errors.

There are many existing tools and frameworks that can be used to build a chatbot, which can help reduce development time and costs. It's important to carefully evaluate these tools and choose ones that fit within the constraints and are aligned with the chatbot's purpose and goals.

The target audience for the chatbot may have limitations that should be considered when building its features. For example, if the chatbot is aimed at users with visual impairments, it may need to include features such as voice commands or text-to-speech functionality.

3.4 Design Flow:

A well-structured flow is the basis of a successful chatbot strategy, as it determines the user journey and interaction with your bot and the course of the conversation, from start to finish.

In this article, we will talk about the strategy of building flows and the best practices to keep in mind when designing your chatbot.

The chatbot flows are the sequential actions that your bot will perform when activated by a trigger. If you are new to chatbots, it is essential to understand that a chatbot flow is composed of three key ingredients: actions, triggers and filters.

An action is literally anything that your bot can do, such as sending a message, sending images or videos or even presenting choices to users. There are many actions your bot can perform with Flow XO, depending on your bot’s objectives.

A trigger is the user input that will lead the bot to take action. The most common triggers are usually messaging triggers, but Flow XO provides different types of triggers that might fit better to your flow.

The filters, however, will select which trigger and action should be executed. It simply determines if a trigger or action should be activated at a certain time. For example, an action may have a filter that will only allow users to continue the conversation if they answered the previous question with “Yes”.

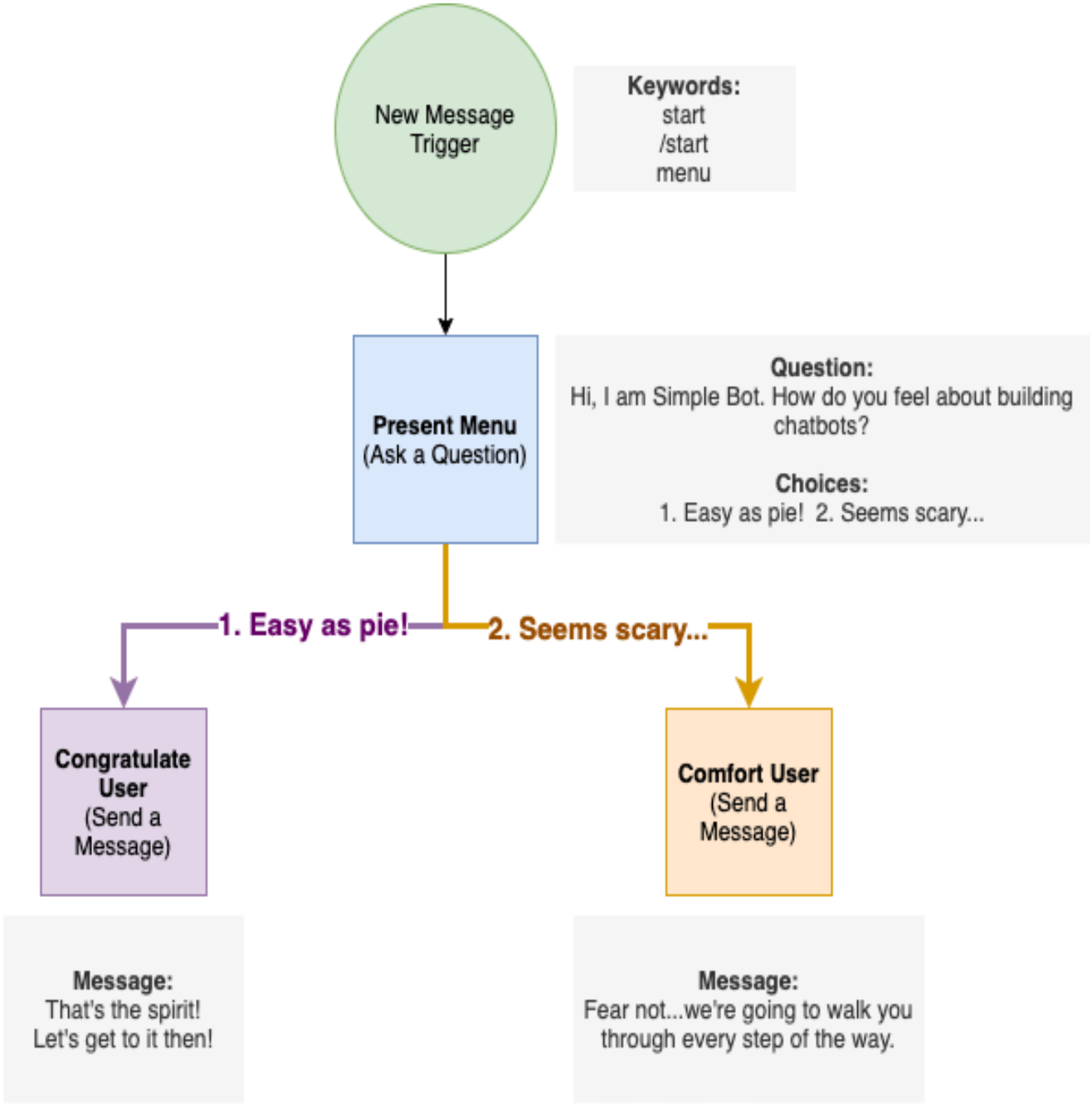


Figure 3.1 Simple flow building

Building Your Bot-

Flow XO can structure bots in different ways, depending on the complexity of the conversation and the overall goals of your chatbot.

One of the first things you need to do is determine if your bot will perform various tasks, called multi-purpose bots, or if it will only follow a simple linear task, called single-purpose bots.

The single-purpose bots are likely to have the main flow that runs the first time a user interacts with your bot. So, for example, if your bot has simple triggers, actions and filters, you might have a single purpose bot with a single flow.

The single flow works more as a funnel than an actual conversation, as most users will receive the messages in sequence. However, a single flow can also cycle back to previous parts of the conversation or integrate with CRM systems, Google Sheets or even messaging channels to engage users and make the flow more sophisticated.

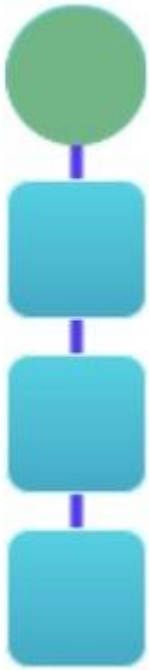


Figure 3.2 Single-purpose bot with single flow

If your conversation has a lot of branching, you will most likely need to use multiple flows to implement your bot, so you will need a single-purpose bot with a multi flow. This kind of flow is built by using one flow to trigger another flow at specific points in the conversation.

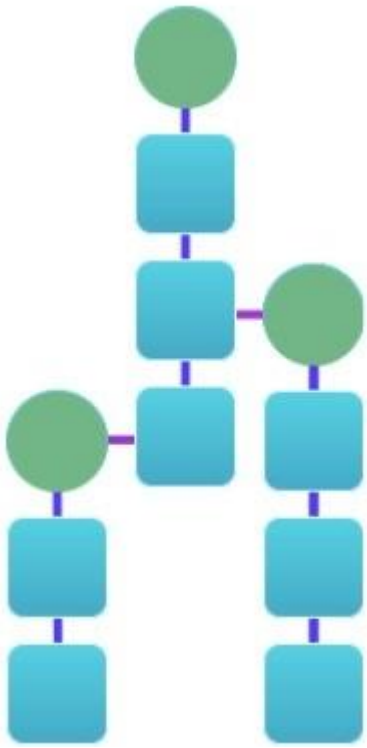


Figure 3.3 Single-purpose bot with multi flow

An example of a single-purpose, multi-flow bot is when a bot assists the customers in making a decision process, such as helping clients see which products will best meet their needs.

On the other hand, the multi-purpose bots accomplish various user tasks, and the bot will need to split into multiple flows. Therefore, your bot presents a “main menu” to users during their first interaction and frequently at the close of each task the bot performs.

You can create different types of menus with multi-purpose bots, such as main menu flow, automated menus, and Pure Natural Language Processing (NLP) menu. If you want to dive deeper into multi-purpose bot menus, you can check the Flow XO support page.

Sharing flows allows you to share one of your flows with other users. For example, you can build a flow for a client and share it with them to configure in their own Flow XO account. In addition, it allows you to create a library of reusable flows. Flow sharing also helps the support team to assist you if you have any issues with your bot.

The use of metadata in flows describes optional extra data that may be sent into a flow. You can also use attributes in flows to store information and access it later. For example, attributes are beneficial if you want your chatbot to remember the user, complete an order, or give a personal greeting.

You can also trigger flows from other flows, which can be very helpful to organize your conversations into multiple flows, and it eliminates the need for filters in each step of your flow.

Flow building offers endless possibilities and mastering this art is key to create a bot with a natural and engaging conversation.

Flow XO is our automation platform that provides a seamless way to create flows that connect things that happen with things you want to do with your bot.

3.5 Design selection:

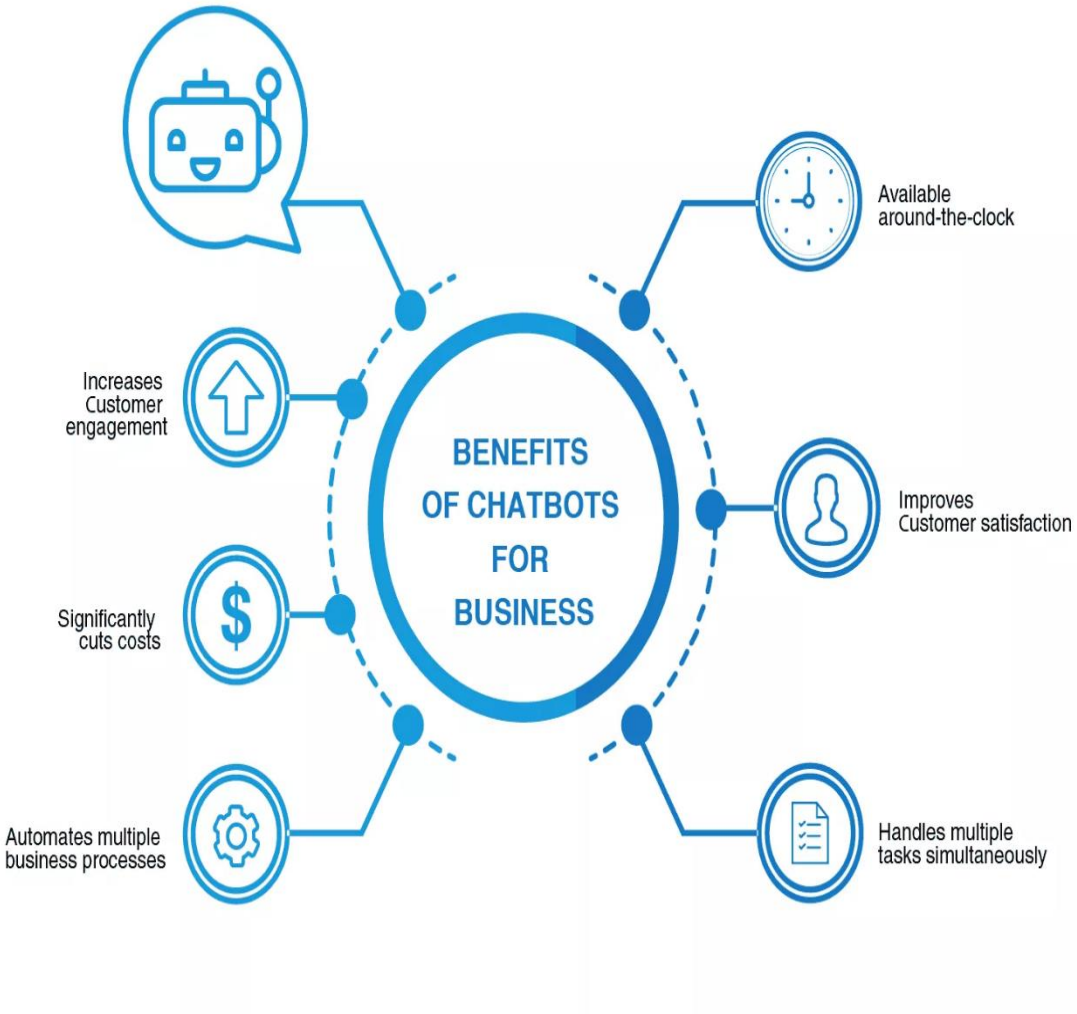


Figure 3.4 Benefits of Chatbots for Business

A good chatbot design has a deeper impact on different business functions such as:

Consistent customer experience – Well-designed chatbots can help you to anticipate customer needs and respond quickly in a personalized way that increases satisfaction. Hence a better customer experience can help to establish a better brand.

Increased customer engagement – Businesses realize how important it is to keep their customers engaged. Intelligently designed chatbots to engage the customers by understanding their intent and providing relevant answers.

Higher lead generation – When your bots are able to provide guidance and tips during customer interaction for successful conversion increases your lead generation.

3.6 Implementation plan/methodology:

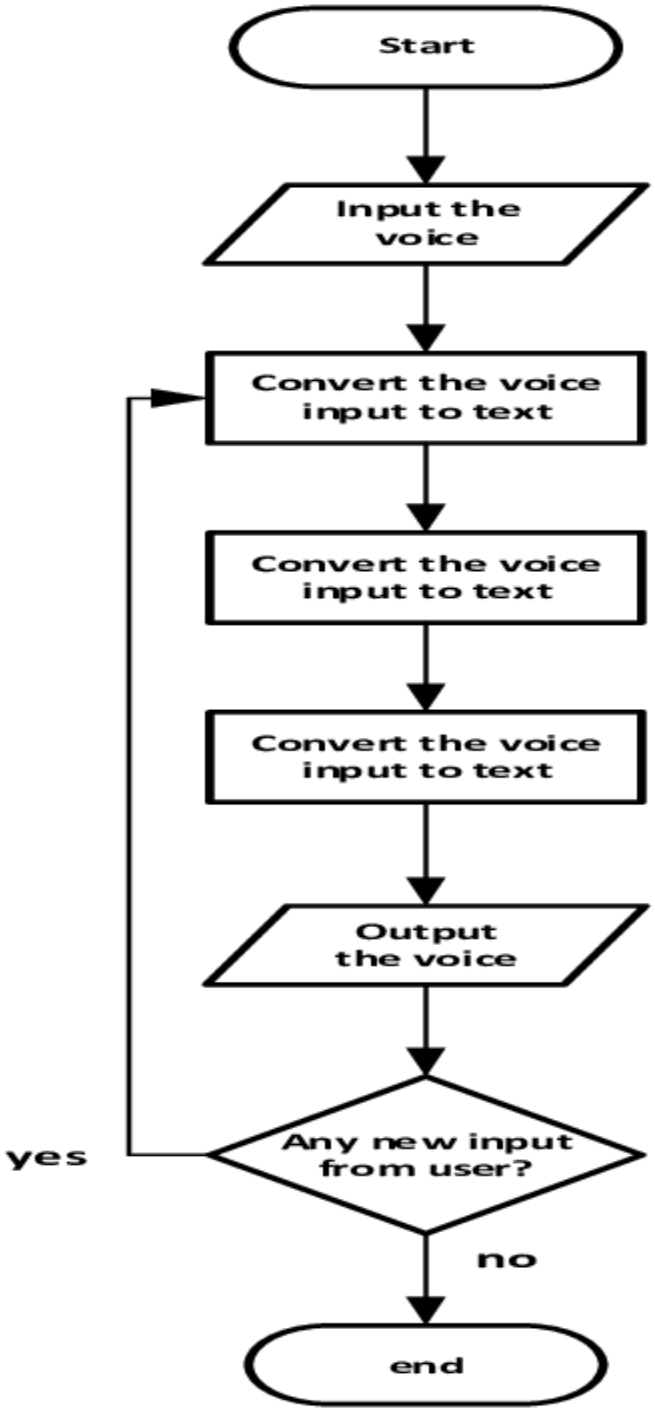


Figure 3.5 Flowchart

Algorithm-

Define the purpose and scope of the chatbot:

Before starting to build the chatbot, it's important to have a clear understanding of why you're building it, what problem it will solve, and what its limitations will be. This will help you determine the appropriate features and functionality for your chatbot.

Determine the platform or tool you will use to build the chatbot:

There are several platforms and tools available to build chatbots, including Facebook Messenger, Slack, WhatsApp, and more. You'll need to choose the platform that best fits your chatbot's purpose and target audience. You can also consider using chatbot development frameworks such as Dialogflow, Botpress, Rasa, and more.

Choose the natural language processing (NLP) engine or service you will use to process user input:

NLP is the key component of a chatbot that enables it to understand and interpret user input. There are several NLP engines and services available, such as Dialogflow, Wit.ai, IBM Watson, and more. You'll need to choose the NLP engine that best fits your chatbot's needs and abilities.

Decide on the type of chatbot you want to build:

There are different types of chatbots, including rule-based, AI-based, and hybrid chatbots. Rule-based chatbots use a set of predefined rules to generate responses to user input. AI-based chatbots, on the other hand, use machine learning algorithms to learn from user interactions and improve their responses over time. Hybrid chatbots combine both rule-based and AI-based approaches.

Develop a list of frequently asked questions (FAQs) or conversation topics:

To get started with building your chatbot, it's helpful to create a list of common questions or conversation topics that your chatbot will be able to handle. This will give you a starting point for building out your chatbot's conversation flow.

Create a database of responses to the FAQs or conversation topics:

Once you have your list of FAQs or conversation topics, you'll need to create a database of responses to each one. For rule-based chatbots, this will involve writing out a set of pre-defined responses to each question or topic. For AI-based chatbots, you'll need to provide examples of user input and the corresponding responses, which the chatbot can use to learn and improve over time.

Write the code for the chatbot's dialogue management system:

The dialogue management system is the component of the chatbot that determines the appropriate response to user input. For rule-based chatbots, this will involve creating a set of if/else statements or switch/case statements to map user input to the appropriate response. For AI-based chatbots, this will involve training the chatbot's machine learning algorithms to recognize patterns in user input and generate appropriate responses.

Test the chatbot to ensure that it can handle a variety of inputs and provide accurate responses:

Once you have your chatbot up and running, it's important to test it thoroughly to ensure that it can handle a variety of inputs and provide accurate responses. You may need to tweak your dialogue management system or provide additional training data to improve your chatbot's performance.

Deploy the chatbot on your chosen platform or tool:

Once you're happy with your chatbot's performance, you can deploy it on your chosen platform or tool. This will typically involve integrating your chatbot with the platform's API and configuring any necessary settings.

Monitor and analyze the chatbot's performance and make adjustments as necessary:

After deploying your chatbot, it's important to monitor its performance and make adjustments as necessary. You can use analytics tools to track metrics such as user engagement.

Block Diagram-

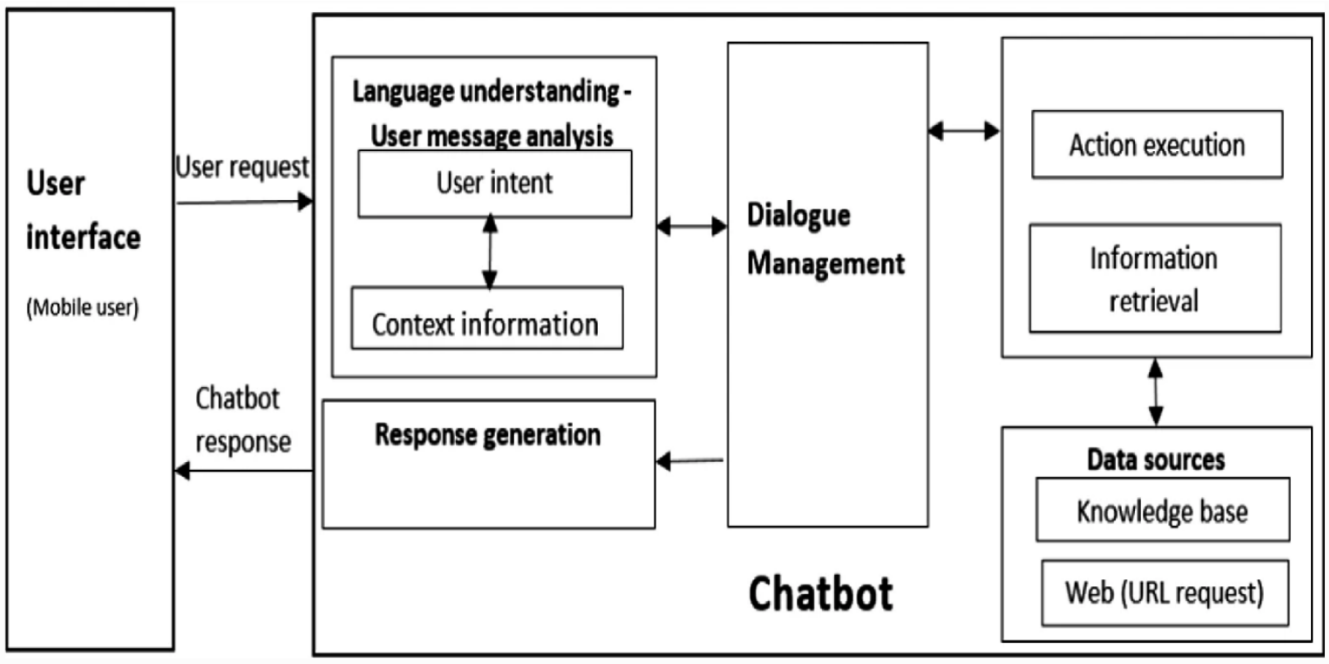


Figure 3.6 Block diagram of chatbot

CHAPTER – 4
RESULTS ANALYSIS AND VALIDATION

4.1 Implementation of Solution:

Implementing a chatbot using natural language processing (NLP) involves several steps. Here's an overview of the process:

- Define the problem: First, you need to define the problem that the chatbot is meant to solve. This will help you determine what type of chatbot you need to build and what kind of functionality it should have.
- Choose a platform: There are many platforms available for building chatbots using NLP.
- Gather training data: Chatbot will need training data to learn how to respond to user input. This data can come from a variety of sources, such as chat logs, customer service transcripts, or public datasets.
- Preprocess the data: Once you have your training data, you'll need to preprocess it to prepare it for use with your chatbot platform. This may involve cleaning the data, formatting it in a specific way, and selecting relevant features.
- Train the model: Next, you'll use your platform to train a machine learning model on the preprocessed data. This will teach the model how to recognize user input and generate appropriate responses.
- Test the model: Once your model is trained, you'll need to test it to ensure it's working properly. You can do this by inputting sample questions and verifying that the chatbot responds appropriately.
- Deploy the chatbot: Once you're confident in your chatbot's performance, you can deploy it to your desired platform, such as a website or messaging app.
- Monitor and update: Finally, it's important to monitor your chatbot's performance over time and update it as necessary. This may involve adding new training data, improving the model's accuracy, or adding new functionality.

Header Files in the code with libraries to download

There are several libraries that are commonly used in chatbot development using natural language processing (NLP). Some of the most popular ones are:

- Natural Language Toolkit (NLTK): This is a popular Python library for NLP, which provides tools for tokenization, stemming, and other text processing tasks.

- Spacy: This is another popular Python library for NLP, which provides a wide range of features such as tokenization, part-of-speech tagging, named entity recognition, and dependency parsing.

- TensorFlow: This is a popular open-source machine learning library that can be used for building and training chatbots with deep learning techniques.

- PyTorch: This is another open-source machine learning library that can be used for building and training chatbots with deep learning techniques.

- Dialogflow: This is a cloud-based platform for building chatbots using NLP, which provides pre-built agents, natural language understanding, and conversation management tools

```
import io
import random
import string # to process standard python strings
import warnings
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import warnings
warnings.filterwarnings('ignore')

import nltk
from nltk.stem import WordNetLemmatizer
nltk.download('popular', quiet=True) # for downloading packages
```

Figure 4.1 Header Files in the code

Reading, Tokenization and Preprocessing of data

```
# uncomment the following only the first time
nltk.download('punkt') # first-time use only
nltk.download('wordnet') # first-time use only


#Reading in the corpus
with open('iml.txt','r', encoding='utf8', errors ='ignore') as fin:
    raw = fin.read().lower()

#TOkenisation
sent_tokens = nltk.sent_tokenize(raw)# converts to list of sentences
word_tokens = nltk.word_tokenize(raw)# converts to list of words

# Preprocessing
lemmer = WordNetLemmatizer()
def LemTokens(tokens):
    return [lemmer.lemmatize(token) for token in tokens]
remove_punct_dict = dict((ord(punct), None) for punct in string.punctuation)
def LemNormalize(text):
    return LemTokens(nltk.word_tokenize(text.lower().translate(remove_punct_dict)))


# Keyword Matching
GREETING_INPUTS = ("hello", "hi", "greetings", "sup", "what's up","hey",)
GREETING_RESPONSES = ["hi", "hey", "*nods*", "hi there", "hello", "I am glad! You are talking to me"]

def greeting(sentence):
    """If user's input is a greeting, return a greeting response"""
    for word in sentence.split():
        if word.lower() in GREETING_INPUTS:
            return random.choice(GREETING_RESPONSES)
```

Figure 4.2 Reading, Tokenization and Preprocessing of data

Reading, tokenization, and preprocessing of data are essential steps in developing a chatbot using natural language processing (NLP) techniques.

- Reading: This step involves gathering the data that will be used to train the chatbot. The data can come from various sources, such as customer service transcripts, chat logs, or publicly available datasets. The data may be stored in various formats, such as text files or databases. The goal of this step is to obtain a dataset that is representative of the language and topics the chatbot will encounter.

- Tokenization: Once the data is obtained, the next step is to tokenize it. Tokenization is the process of breaking the text into individual tokens or words. This is important because it enables the chatbot to process the input text more efficiently. There are various ways to tokenize text, such as using regular expressions, whitespace, or specialized NLP libraries.

- Preprocessing: The final step is preprocessing the data, which involves cleaning and formatting the data to prepare it for use with the chatbot. This may include removing punctuation and special characters, converting text to lowercase, and removing stop words (common words that do not carry much meaning, such as "the" or "a"). Other preprocessing steps may involve stemming or lemmatization, which involves reducing words to their root forms to enable the chatbot to recognize related words (e.g., "run" and "running").

Overall, reading, tokenization, and preprocessing are critical steps in building a chatbot using NLP. These steps help to ensure that the chatbot is trained on a representative dataset and can process user input effectively.
general architecture of a chatbot using NLP.

Architecture of Chatbot

The chatbot architecture typically consists of three components:

- NLP engine: This component is responsible for processing user input, understanding its meaning, and extracting relevant information to generate an appropriate response. This component involves different NLP techniques such as tokenization, stemming, lemmatization, and part-of-speech tagging.

- Dialog Management: This component manages the dialog flow between the user and the chatbot. It includes keeping track of the context of the conversation, detecting user intent, and deciding the next action the chatbot should take.

- Integration with external services: This component handles external services such as databases, APIs, or third-party services to provide additional functionality to the chatbot.

Overall, the chatbot architecture is highly modular and flexible, allowing for easy customization and integration with various platforms and technologies.

Code Explanation

This code is an implementation of a chatbot using Natural Language Processing (NLP) techniques in Python. The chatbot reads in a corpus of text, tokenizes it, performs preprocessing, and then generates responses to user input based on a keyword matching approach and cosine similarity.
Here is an overview of the code:

- The code starts by importing the necessary libraries, including NumPy, scikit-learn, and NLTK.

- It then reads in a corpus of text from a file using the open () function and stores it in a variable called "raw". The "raw" text is then converted into a list of sentences using the sent_tokenize() function from the NLTK library, and a list of words using the word_tokenize() function.

- Next, the code defines a LemTokens() function that takes a list of tokens as input and returns a list of lemmatized tokens (i.e., the base form of each word). It also defines a LemNormalize() function that takes a text string as input, tokenizes it, removes punctuation, converts it to lowercase, and then returns a list of lemmatized tokens.

- The code then defines a set of greeting inputs and responses, and a greeting() function that takes a sentence as input and returns a random greeting response if the sentence contains a greeting input.

- The response() function is defined next, which takes a user input as an argument and generates a response based on cosine similarity between the input and the corpus sentences. It tokenizes the user input, preprocesses it using the LemNormalize() function, and then converts it into a TF-IDF matrix using the TfidfVectorizer() function from scikit-learn. It then calculates cosine similarity between the input and each sentence in the corpus, selects the sentence with the highest similarity, and returns it as the response.

- Finally, the code enters a loop that prompts the user for input and generates a response until the user types "bye" to exit. The loop checks for greeting inputs and responds accordingly, or uses the response() function to generate a response based on the user input.

Overall, this code provides a basic framework for building a chatbot using NLP techniques, including tokenization, preprocessing, and cosine similarity. However, it is important to note that this is a simple implementation, and more complex chatbots may require additional techniques such as named entity recognition, sentiment analysis, and machine learning.

```
# Generating response
def response(user_response)
    robo_response=''
    sent_tokens.append(user_response)
    TfidfVec = TfidfVectorizer(tokenizer=LemNormalize, stop_words='english')
    tfidf = TfidfVec.fit_transform(sent_tokens)
    vals = cosine_similarity(tfidf[-1], tfidf)
    idx=vals.argsort()[0][-2]
    flat = vals.flatten()
    flat.sort()
    req_tfidf = flat[-2]
    if(req_tfidf==0):
        robo_response=robo_response+"I am sorry! I don't understand you"
        return robo_response
    else:
        robo_response = robo_response+sent_tokens[idx]
        return robo_response


flag=True
print("ROBO: My name is Robo. I will answer your queries about Chatbots. If you want to exit, type Bye!")
```

Loading...

Figure 4.3 Generating Response 1

```
flag=True
print("ROBO: My name is Robo. I will answer your queries about Chatbots. If you want to exit, type Bye!")
while(flag==True):
    user_response = input()
    user_response=user_response.lower()
    if(user_response!='bye'):
        if(user_response=='thanks' or user_response=='thank you' ):
            flag=False
            print("ROBO: You are welcome..")
        else:
            if(greeting(user_response)!=None):
                print("ROBO: "+greeting(user_response))
            else:
                print("ROBO: ",end="")
                print(response(user_response))
                sent_tokens.remove(user_response)
    else:
        flag=False
        print("ROBO: Bye! take care..")
```

Figure 4.4 Generating Response 2

OUTPUT:

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
ROBO: My name is Robo. I will answer your queries about Chatbots. If you want to exit, type Bye!
Hi
ROBO: hey
What is Indian motor law
ROBO: an act to consolidate and amend the law relating to motor vehicles.
what is certificate of regisration
ROBO:document that proves your vehicle is registered with the Regional Transport Office (RTO) authorities.
What is private service vehicle
ROBO: private service vehicle means a bus being used as part of an Excursion or Tour.
```

Figure 4.6 Output Screenshot 1

```
ROBO: My name is Robo. I will answer your queries about Chatbots. If you want to exit, type Bye!
where is Chandigarh University
ROBO: I am sorry! I don't understand you
bye
ROBO: Bye! take care..
```

Figure 4.7 output Screenshot 2

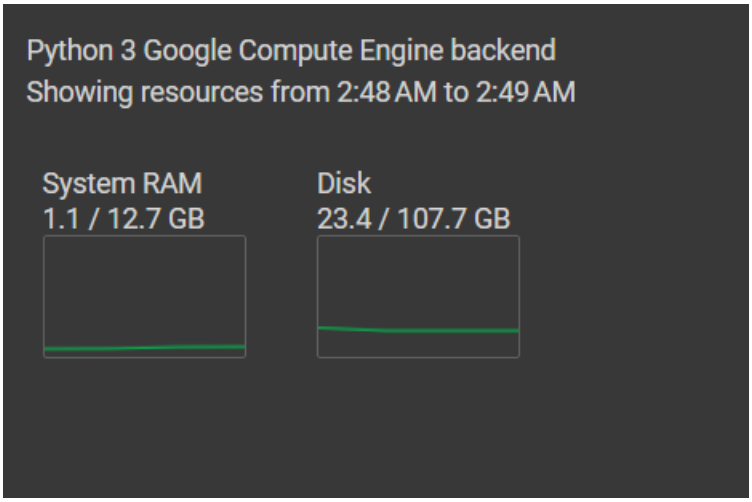


Figure 4.8 System Resources Usage during runtime of the program

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion:

In this paper, we have presented a smart question answering system for the legal domain that uses a .txt file as a knowledge source. Our system can process the input file and extract relevant information to answer user queries. We have used natural language processing and machine learning techniques to analyze the input file and generate answers. We have also implemented a user interface that allows users to interact with the system easily and intuitively. We have demonstrated the effectiveness of our system on various types of questions and compared it with existing systems. Our system aims to provide direct and accurate answers to users who need legal assistance or information. We believe that our system can be a useful tool for legal professionals and researchers, as well as for general users who are interested in the legal domain. As future work, we plan to extend our system to handle more complex and diverse questions, and to incorporate more knowledge sources from different domains.

5.2 Future Scope:

The future of chatbots is transforming the way businesses interact with their customers. From handling customer inquiries and offering real-time support to providing personalized product recommendations, chatbots are becoming increasingly important for all types of businesses in the digital age.

With continued advancements in AI and machine learning technology, it’s clear that chatbot usage is only going to continue to increase across industries.

At Onix, we have more than 20 years of experience investigating and developing chatbot technology for our clients, so we have decided to share our view on top trends.

In this blog post, we'll look at 10 key trends surrounding chatbots and offer insights into the size of this rapidly growing market segment from a global perspective.

The chatbot industry is projected to reach a market size of $3.62 billion by 2030, with an annual growth rate of 23.9%. This speaks to the increasing popularity of chatbots and their potential as powerful customer service tools. This trend has been primarily driven by the advancements in artificial intelligence (AI) and machine learning (ML) technologies, which have allowed for more sophisticated chatbot solutions. As a result, businesses increasingly leverage AI-driven chatbots as part of their customer service operations.

The healthcare industry is currently one of the leading sectors for chatbot adoption, with 43% of companies using them for customer service. Additionally, 33% of businesses in the manufacturing industry and 28% of companies in the retail sector are leveraging chatbots as a tool for customer support. Furthermore, research from Oracle also found that 29% of banking and finance-related organizations were deploying artificial intelligence-powered chatbots to improve user experience. For other industries such as media, e-commerce, hospitality, and travel, there was between a 13-15% usage rate. As we can see, more and more companies are adopting chatbot technology and making use of it. Let’s now dive deeper into the chatbot industry trends we might expect in this sphere for 2023 and beyond.

Modern chatbots are evolving into what scientists initially wanted them to be: they actually learn over time. This learning occurs mainly through human interaction, but that's not the only option. Chatbots also collect so-called training data and can be connected to open-source data (like Wiki QA Corpus or Ubuntu Dialogue Corpus) to create a fuller picture. Chatbots use this during a live chat as a reference. The more data that comes in, the more capable chatbots can process and understand. They work smoothly and efficiently, their reactions become increasingly personalized, and operational time becomes shorter. It is quite likely that AI chatbots will become fully capable of assisting with the user's needs at nearly every stage of the customer experience. This means much less generic and useless information is communicated.

As technology advances, it has become increasingly commonplace to see chatbot utilization not just in customer service industries but in various businesses. By 2023, market analysts expect chatbots to be integral to every industry as consumers continue to expect 24/7 customer service. But instead of simple bots offering pre-programmed answers or scripts, these modern chatbots are expected to use natural language processing and hold conversations that closely mimic human interaction. With the expected growth of AI-driven digital assistants, rise in consumer expectations and need for automation efficiency, businesses' average conversational AI investment is set to reach $18.4 billion by 2026. It's clear that this trend is only likely to grow and develop in the coming years and, with continued advancements in technology, could offer incredible potential for businesses and consumers alike.

REFERENCES

1. Alepis, E., & Virvou, M. (2011). Automatic generation of emotions in tutoring agents for affective e-learning in medical education. Expert Systems with Applications, 38(8): 9840–9847.

2. Ashok, G., Brian, C., Mithun, K., Shanu, S., Abhinaya, S., & Bryan, W. (2015). Using Watson for Enhancing Human-Computer Co-Creativity. AAAI Symposium: 22–29.

3. Avalverde, D. (2019). A Brief History of Chatbots. Perception, Control, Cognition. Retrieved March 9, 2019 from: https://pcc.cs.byu.edu/2018/03/26/a-brief-history-of-chatbots/

4. Ayedoun, E., Hayashi, Y., & Seta, K. (2015). A Conversational Agent to Encourage Willingness to Communicate in the Context of English as a Foreign Language. Procedia Computer Science, 60(1): 1433–1442.

5. Ben Mimoun, Mohammed Slim, & Poncin, I. (2015). A valued agent: How ECAs affect website customers' satisfaction and behaviors. Journal of Retailing and Consumer Services, 26: 70– 82.

6. Chatbot Magazine (2019). A Visual History of Chatbots. Retrieved March 9, 2019 from: https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2

7. Colace, F., De Santo, M., Lombardi, M., Pascale, L., Pietrosanto, A. (2018). Chatbot for E-Learning: A Cases Study. International Journal of Mechanical Engineering and Robotics Research Vol. 7, No. 5, September.

8. Egencia (2018). What is a Chatbot and How does it work? Retrieved March 9, 2019 from: https://www.youtube.com/watch?v=38sL6pADCog

9. Hattie, J. (2012). Visible learning for teachers: Maximizing impact on learning: Routledge. https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2

10. Lip ko, H. (2018). Meet Jill Watson: Georgia Tech's first AI teaching assistant. Retrieved on March 9, 2019 from: https://pe.gatech.edu/blog/meet-jill-watson-georgia-techs-first-ai-teaching-assistant.

11. Maruti Techlabs. (2018). Why can chatbots replace Mobile Apps immediately? Retrieved March 9, 2019 from: https://www.marutitech.com/why-can-chatbots-replace-mobile-apps-immediately/

12. Nguyen, M. (2017). How artificial intelligence & machine learning produced robots we can talk to. Business Insider. Retrieved March 9, 2019 from: https://www.businessinsider.com/what-is-chatbot-talking-ai-robot-chat-simulators-2017-10

13. Simplilearn (2018). Machine Learning Basics. Retrieved March 9, 2019 from: https://www.youtube.com/watch?v=ukzFI9rgwfU

14. Sproutsocial.com (2018). A complete Guide to Chatbots in 2018. Retrieved March 9, 2019 from: https://sproutsocial.com/insights/topics/chatbots/

15. V Soft Consulting. (2019). 7 of the best Language-learning Chatbot Apps. Retrieved March 9, 2019 from: https://blog.vsoftconsulting.com/blog/7-of-the-best-language-learning-chatbot-apps

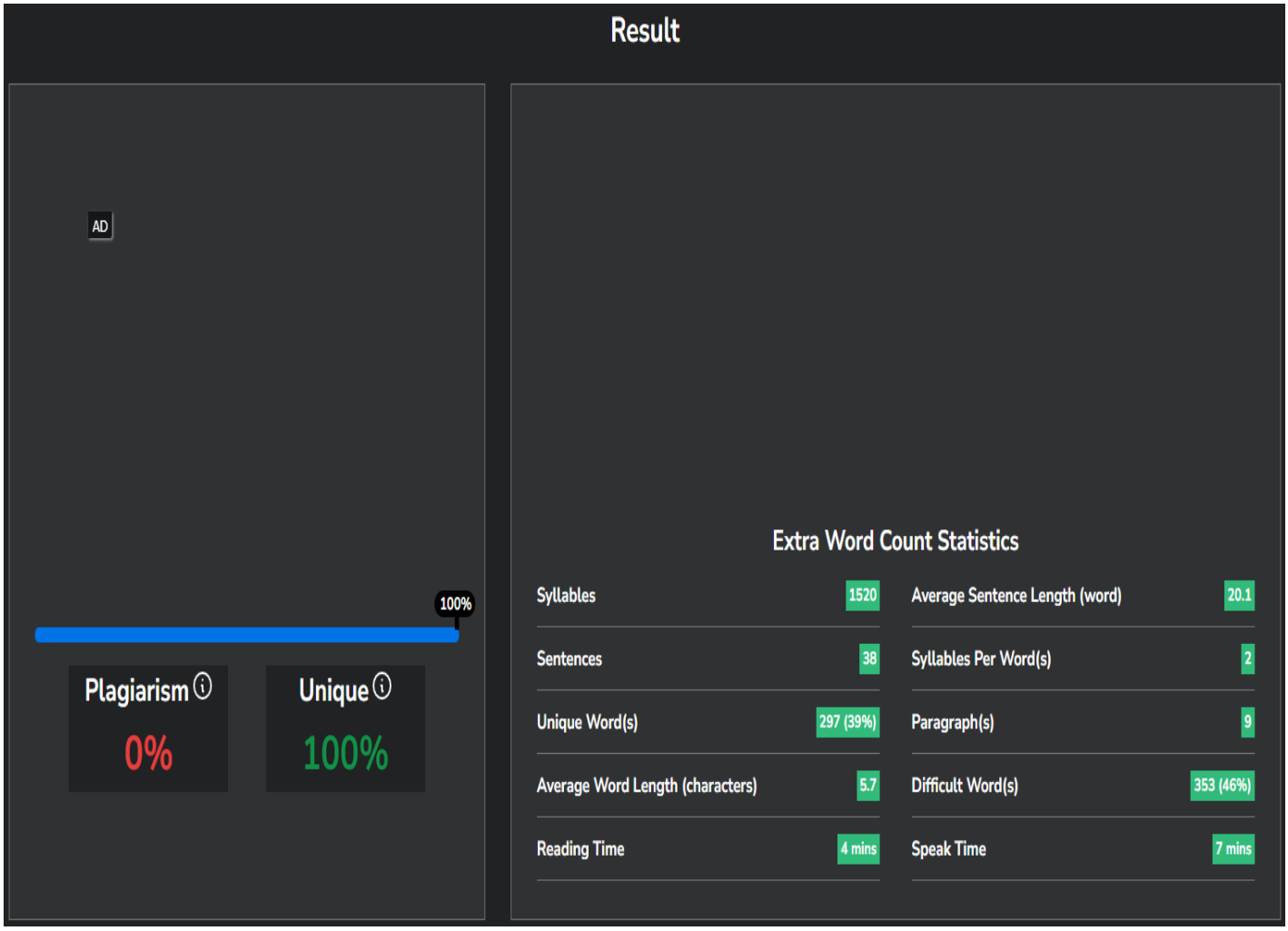
16. Wikipedia (2019). Chatbot. Retrieved March 9, 2019 from: https://en.wikipedia.org/wiki/Chatbot

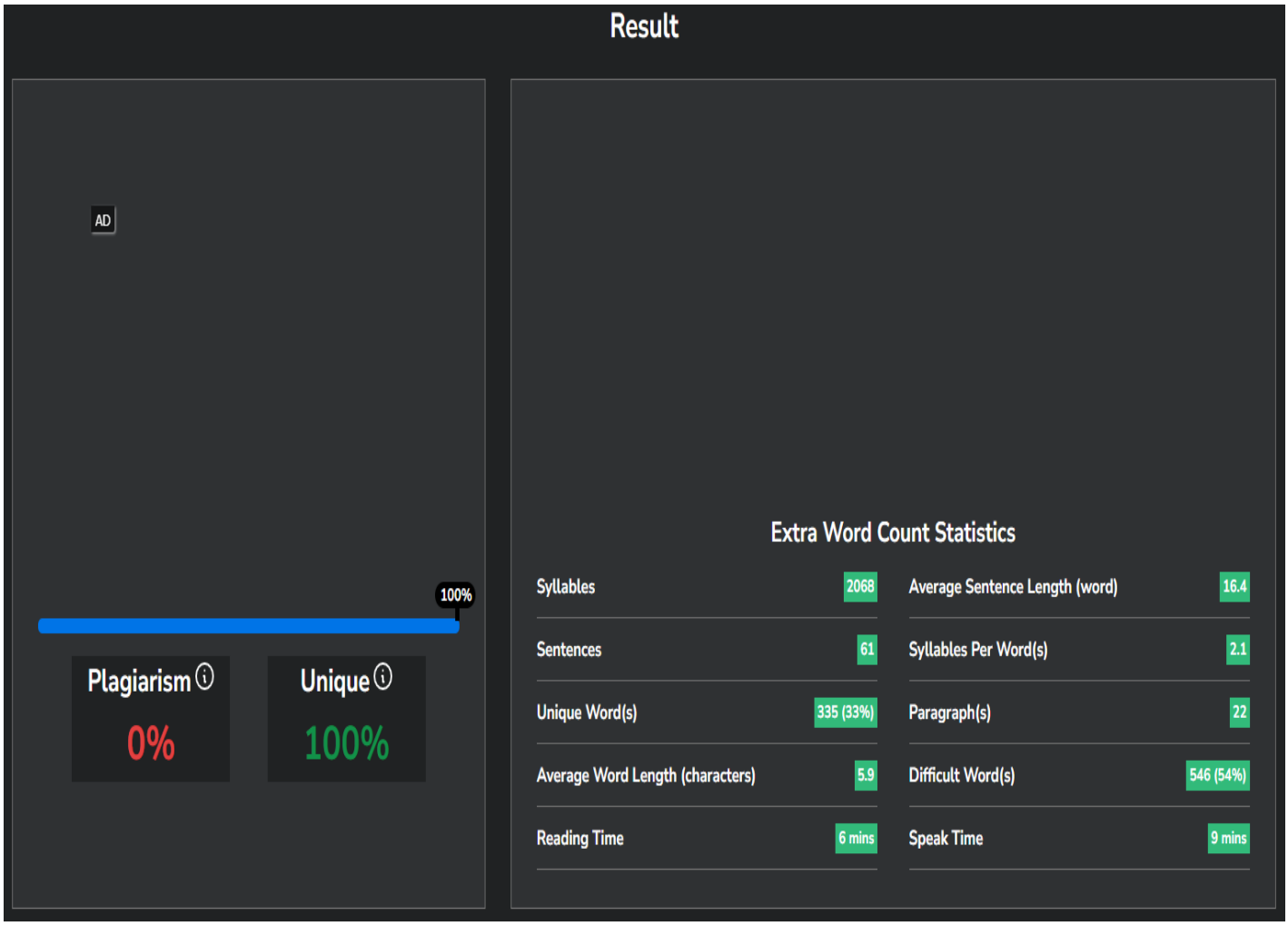
17. Wikipedia (2019). Siri. Retrieved March 9, 2019 from: https://en.wikipedia.org/wiki/Siri#Features_and_options

18. Winkler, R., Söllner, M. (2018): Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis. In: Academy of Management Annual Meeting (AOM). Chicago, USA.

APPENDIX

1. Plagiarism Report:




Result
AD
100%
Plagiarism
0%
Unique
100%
Extra Word Count Statistics
Syllables 2068
Average Sentence Length (word) 16.4
Sentences 61
Syllables Per Word(s) 2.1
Unique Word(s) 335 (33%)
Paragraph(s) 22
Average Word Length (characters) 5.9
Difficult Word(s) 546 (54%)
Reading Time 6 mins
Speak Time 9 mins

2. Design Checklist:

Here's a design checklist for a legal question-answering chatbot using NLP:

- **Determine the chatbot's scope:** Decide on the specific legal domain(s) the chatbot will cover. Will it be focused on corporate law, criminal law, intellectual property law, or another area of law?
- **Identify the user persona(s):** Determine the chatbot's target audience, such as lawyers, law students, or individuals with legal questions.
- **Collect data and create a knowledge base:** Gather relevant legal documents, cases, and statutes to use as the basis for the chatbot's knowledge base. This can include commonly asked legal questions and their answers.
- **Develop the chatbot's language processing capabilities:** Use natural language processing (NLP) to enable the chatbot to understand and respond to user queries in a conversational and intuitive way.
- **Test the chatbot's accuracy:** Test the chatbot's responses against a set of legal questions to ensure it provides accurate and relevant answers.
- **Ensure compliance with legal regulations:** Ensure that the chatbot complies with relevant legal regulations and ethical standards, such as attorney-client privilege and data privacy laws
- **Design a user-friendly interface:** Create a user-friendly chat interface that allows users to easily enter their questions and receive accurate responses.
- **Incorporate feedback mechanisms:** Implement a feedback mechanism to allow users to rate the chatbot's performance and provide feedback on areas for improvement.
- **Monitor and update the chatbot:** Regularly monitor the chatbot's performance and update its knowledge base to ensure it stays up to date with changes in legal regulations and standards.
- **Provide customer support:** Provide users with a way to contact customer support in case they encounter any issues or have questions that the chatbot cannot answer.

USER MANUAL

Welcome to the User Manual for Chatbot for legal question answering using NLP. This chatbot has been developed to provide users with quick and accurate answers to legal questions using natural language processing (NLP) technology. This User Manual will guide you on how to use the chatbot.

- Accessing the Chatbot:
  1. To access the Chatbot, you need to visit the website or application where the Chatbot is available.
  2. Once you are on the website or application, you can initiate a conversation with the Chatbot by clicking on the Chatbot icon or typing in a message.

- Asking Legal Questions:
  1. To ask a legal question, type in your query in the chat window.
  2. The chatbot will analyze your query and provide an accurate response based on the information it has been trained on.
  3. If the Chatbot does not understand your query, it will prompt you to rephrase your question.

- Interacting with the Chatbot:
  1. The Chatbot will respond to your queries in a conversational manner, just like a human lawyer would.
  2. You can ask follow-up questions or request clarification on any of the answers provided by the Chatbot.
  3. The Chatbot will provide references to legal sources, if available, to support its responses.

- Limitations of the Chatbot:
  1. The Chatbot is designed to answer legal questions based on the information it has been trained on.
  2. The Chatbot may not be able to provide a response to all legal questions, particularly those that are highly complex or require specific legal expertise.
  3. The Chatbot should not be relied upon as a substitute for professional legal advice.

- Confidentiality:
    1. The Chatbot is designed to maintain user confidentiality.
    2. The Chatbot will not share any personal information or details of the conversation with third parties.

- Reporting Issues:
    1. If you encounter any issues or have any feedback regarding the Chatbot, please contact the developers through the provided support channels.

- Accuracy and Reliability:
    1. The Chatbot uses natural language processing (NLP) technology to analyze user queries and provide responses.
    2. The Chatbot has been trained on a large corpus of legal documents and case law to ensure accuracy and reliability of responses.
    3. However, the Chatbot's responses are based on the information it has been trained on and may not take into account specific nuances or exceptions in certain legal situations.

- Use as a Tool for Legal Research:
    1. The Chatbot can be used as a tool for legal research, as it provides references to legal sources to support its responses.
    2. Users can access these references and conduct further research to gain a deeper understanding of the legal issue at hand.

- Understanding Legal Advice:
    1. The Chatbot is designed to provide information and guidance on legal issues, but it should not be relied upon as a substitute for professional legal advice.
    2. Legal advice takes into account specific facts and circumstances of a particular case, and only a licensed attorney can provide legal advice.

- Continuous Learning and Improvement:
    1. The Chatbot is continuously learning and improving through user feedback and updates to its training data.
    2. Users can provide feedback

Thank you for using the Chatbot for legal question answering using NLP. We hope this User Manual has been helpful in guiding you on how to use the Chatbot effectively.