

Article

# Internet Wizard for Enhancing Open-Domain Question-Answering Chatbot Knowledge Base in Education

Kovan Mzwri <sup>1,\*</sup>  and Márta Turcsányi-Szabo <sup>2</sup><sup>1</sup> Doctoral School of Informatics, Eötvös Loránd University, Pázmány Péter stny. 1/C, 1117 Budapest, Hungary<sup>2</sup> Department of Media & Educational Technology, Faculty of Informatics, Eötvös Loránd University, Pázmány Péter stny. 1/C, 1117 Budapest, Hungary; tsmarta@inf.elte.hu

\* Correspondence: kovan@inf.elte.hu; Tel.: +36-707438561

**Abstract:** Chatbots have gained widespread popularity for their task automation capabilities and consistent availability in various domains, including education. However, their ability to adapt to the continuously evolving and dynamic nature of knowledge is limited. This research investigates the implementation of an internet wizard to enhance the knowledge base of an open-domain question-answering chatbot. The proposed approach leverages search engines, particularly Google, and its features, including feature snippets, knowledge graph, and organic search, in conjunction with data science and natural language models. This mechanism empowers the chatbot to dynamically access the extensive and up-to-date knowledge available on the web, enabling the provision of real time and pertinent answers to user queries sourced from web documents. A pilot study in a higher education context evaluated the chatbot's mechanism and features, confirming its proficiency in generating responses across a broad range of educational and non-educational topics. Positive feedback and high user satisfaction validate these findings. Notably, the chatbot's dynamic feature of retrieving related or follow-up questions from search engines significantly enhances student engagement and facilitates exploration of supplementary information beyond the curriculum.

**Keywords:** chatbots; dynamic knowledge base; internet wizard; search engine integration; Google feature snippets; open-domain question-answering chatbot; knowledge graph; dynamic knowledge acquisition; student engagement; higher education



**Citation:** Mzwri, K.; Turcsányi-Szabo, M. Internet Wizard for Enhancing Open-Domain Question-Answering Chatbot Knowledge Base in Education. *Appl. Sci.* **2023**, *13*, 8114. <https://doi.org/10.3390/app13148114>

Academic Editor: Antonio Sarasa Cabezuelo

Received: 21 June 2023

Revised: 8 July 2023

Accepted: 10 July 2023

Published: 12 July 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Chatbots as an application of human–computer interaction have gained significant importance in the last few years and undergone remarkable developments. The growing interest in chatbots can be attributed to the proliferation of mobile devices over the past decade. As these smart devices, such as smartphones, have become increasingly popular, so too have the applications that run on them, including chatbots. Consequently, chatbots have transformed the ways in which humans interact with technology and have created new opportunities for organizations and businesses to engage with their clients. These programs are designed to perform a wide range of tasks in an automated manner, making them a useful tool in various settings [1]. Chatbots, also called conversational agents, can be defined as conversational or interactive agents that provide an instant response to the user [2,3]. A computer program or artificial intelligence tool that conducts a conversation using auditory or textual methods.

Chatbots, or conversational agents, have been around since the development of the first application in 1960 called ELIZA by Joseph Weizenbaum at MIT. ELIZA used natural language processing (NLP) to recognize patterns and act as a therapist [4]. Other notable examples include ALICE, developed by Richard Wallace in 1995, which uses Artificial Intelligence Markup Language (AIML) for pattern matching to recognize inputs and generate responses [5].

In the past, chatbots were primarily used in the customer service industry, but in recent years, they have become more sophisticated and able to understand and respond to more complex inputs. They are now used in a wide range of industries, such as health-care [6,7], finance [8,9], customer service [10,11], individualized support via intelligent audio device [12], and education such as using chatbots to learn Computer Programming concepts [3,13,14]. Their widespread adoption is anticipated to grow even further in the future [15].

Educational institutions have increasingly adopted chatbots to enhance educational support and provide information to students. They are implemented in various settings, including schools, universities, and online platforms, as mobile web applications. Through chatbots, students can instantly access standardized details, such as course materials [16], practice questions and answers [17,18], evaluation criteria [19,20], assignment due dates, advice [21], and study materials. Implementing such systems can enhance student engagement, reduce the administrative burden on lecturers, and allow for more focus on curriculum development [16]. Chatbots not only help students develop their interaction skills but also assist teaching faculty by automating some of their tasks [22]. Integrating chatbots with education improves connectivity, efficiency, and reduces uncertainty in interactions [23,24], leading to personalized, and result-oriented online learning experiences [16], which are vital for today's educational institutions. Furthermore, chatbots increase the level of support provided to each student, reducing the likelihood of ineffective learning and dropouts [25,26].

Chatbots, whether domain-specific or open-domain, usually undergo a training process that involves teaching them to understand and respond to user inputs in a natural and human-like manner. This is achieved through the use of machine learning techniques, including natural language processing (NLP) and deep learning.

Various natural language processing (NLP) models, including BERT, RoBERTa, GPT-2, and GPT-3, are utilized for developing open-domain chatbots. These models are trained on extensive general-domain text data and fine-tuned on task-specific datasets to adapt them to the chatbot's purpose [27–29]. Commonly used datasets, such as pushshift.io Reddit [30] and Empathetic Dialogues [31], are used to train the weights of a Transformer encoder–decoder, as seen in state-of-the-art chatbots such as Meena [32] and BlenderBot [33]. Moreover, language models such as GPT4 and LaMDA have made significant strides in natural language processing, exhibiting the ability to generate contextually relevant and coherent text.

Nevertheless, these models primarily rely on the information provided within the training datasets and do not incorporate external knowledge sources to enhance their generative capabilities. This static language modeling approach neglects the dynamic nature of the world, where new information continuously emerges. In essence, these models are trained based on data collected during their creation, resulting in static knowledge that fails to adapt to changes. Consequently, the models may encounter difficulties in offering up-to-date or evolving responses, potentially leading to inaccuracies or outdated information in the generated dialogue.

To keep an open-domain chatbot's knowledge base current, it is important to continuously update it with new information. One way to achieve this is through web scraping techniques that gather data from sources such as news websites, social media, and forums. APIs can also be used to access real-time data from external sources. By incorporating new information in real time, chatbots can maintain an up-to-date knowledge base that is responsive to user needs.

The primary objective of this research is to develop a cost-effective mechanism that can handle the ever-changing nature of acquired knowledge and empower a question-answering chatbot to respond to queries from diverse fields. To achieve this objective, the study explores the potential of contemporary internet search tools, especially search engines and their features. The research integrates data science and machine learning algorithms to

retrieve and analyze knowledge from the world wide web, creating a dynamic knowledge base that can be leveraged by an open-domain QA chatbot.

Furthermore, the research investigates the potential of utilizing dynamic-related questions or follow-up questions retrieved from search engines to enhance student engagement with the chatbot. This strategy aims to encourage students to seek additional information about specific queries in a higher education setting, thereby promoting active information seeking. Incorporating dynamic follow-up questions is expected to improve the chatbot's effectiveness in addressing a wide range of queries and promoting active information-seeking among students. To answer the research questions, a pilot study will be conducted to evaluate the effectiveness of the developed chatbot in an educational setting.

This paper is organized as follows: Section 2 introduces important related work. In Section 3, the chatbot architecture including its components, features, and technologies utilized is outlined. Section 4 presents the Pilot Study, including Data Collection and Evaluation Method. The experimental results and discussions are reported in Section 5. Finally, Section 6 concludes the paper, discussing its limitations and suggesting future work.

## 2. Related Work

The field of chatbot development has undergone significant advancements since the inception of the first chatbot, which relied on rule-based knowledge bases with limited scope and simplistic pattern matching. Numerous studies have explored the incorporation of offline and online external knowledge to enhance chatbot knowledge bases.

In the offline approach, knowledge bases have been created using text corpora, such as didactic textbooks, through semi-automated processes such as Automatic Generation of AIML from Text Acquisition (AGATA). This involves human intervention at certain stages [34].

Online approaches have also been investigated to expand chatbot knowledge bases. Researchers have explored knowledge extraction from online communities for tasks such as question-answering and summarization such as a knowledge base for a QA system that answers "how" questions was developed [35]. Additionally, a method was presented to detect question-answer pairs in email conversations for email summarization [36]. Furthermore, online communities such as forums have been utilized for automatic chatbot knowledge acquisition [37].

In the context of the semantic web, chatbot systems leverage structured and linked data to facilitate conversations and perform various tasks such as question answering and FAQs. For instance, SOGO [38] is a semi-automatic social dialogue system that combines task-oriented utterances with conversational strategies to engage users in negotiation. Another example is La Liga [39], a social chatbot designed for the football domain, specifically catering to a diverse range of questions related to the Spanish football league. It uses an NLU block trained to extract intents and associated entities from user questions, obtaining information by making real-time SPARQL queries to the Wikidata knowledge base site. Additionally, KBot [40] utilizes semantic web techniques and linked data, making use of large-scale, publicly available knowledge bases such as DBpedia, Wikidata, and myPersonality.

While these approaches capitalize on the characteristics of their respective corpora and tasks, they are limited by the scope and resources of their specific knowledge bases. Furthermore, although datasets derived from natural or crowdsourced data provide valuable resources for training dialogue generation models, they are static in nature and lack real-time updates or the ability to incorporate evolving information. This can lead to challenges in generating up-to-date and accurate responses, potentially resulting in outdated or inaccurate information.

Notably, other studies [41–43] have explored the possibility of knowledge selection from a small set of knowledge, without employing a retrieval step or search engine, as was performed in this study. It is worth mentioning that the utilization of search engines for machine translation tasks has been shown to produce effective results [44]. The Wizard of

Wikipedia task [45] involves conversations grounded in Wikipedia and utilizes a TFIDF retrieval model to find relevant information from the database. Similarly, our previous study [46] employed the Wikipedia API to retrieve related knowledge from Wikipedia and Canvas LMS and answer user queries. Those studies are perhaps the closest to this research work. However, this research work adopts a more comprehensive approach, leveraging publicly available internet information and combining data science and machine learning models with Google’s search engine features, including feature snippets, knowledge graph, and organic search results, for dynamic chatbot knowledge acquisition, enabling the delivery of up to date and precise responses to user queries. To the best of our knowledge, no published literature comprehensively elucidates the utilization of search engines, specifically Google, and its associated features, in conjunction with data science and machine learning models, for the purpose of dynamic knowledge acquisition in question- answering chatbots.

### 3. Chatbot Architecture

This section describes the architecture of the developed question-answering chatbot.

Figure 1 illustrates the structure of the chatbot system, encompassing its components, technologies employed, and the interconnectedness that enables a seamless conversational experience for the user. Overall, architectures for chatbots are dependent upon their specific use cases and functionality requirements.

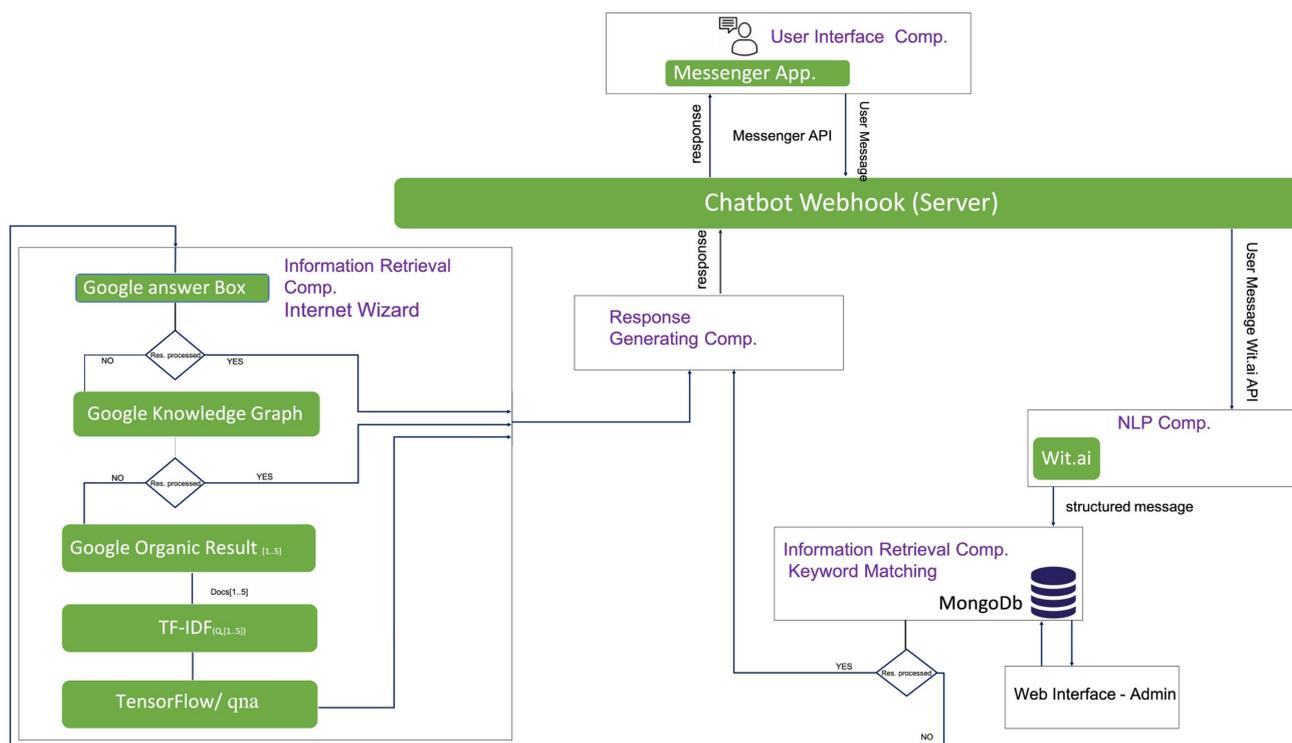


Figure 1. Chatbot architecture.

The type of chatbot developed in this study is a question-answering (QA) chatbot named Kucko Version 2. A question-answering (QA) chatbot is a type of chatbot designed to provide answers to user questions in natural language. The main goal of a QA chatbot is to provide accurate and relevant information to the user in real time. Natural language processing (NLP) and machine learning techniques are typically used by QA chatbots to understand the user’s question and retrieve the most relevant answer from its knowledge base. The knowledge base can be a structured database, unstructured text, or a combination of both. The main challenge of the question-answering chatbot is its limited knowledge base [46]. The architecture of the chatbot consists of a combination of various technical and

functional components that work together to provide an appropriate and up-to-date response to user queries. These components work together to enable a chatbot to understand user input, retrieve information from its knowledge base, and generate an appropriate response in real time. In general, the developed chatbot is composed of four components: user interface, natural language processing, information retrieval, and response generation. In each of those components, a variety of technologies have been used emphasizing their accessibility and public availability. As the primary part of this research, information retrieval focuses on the mechanism for retrieving dynamic knowledge from the world wide web which we refer to as the Internet Wizard.

### 3.1. User Interface

User interface is the first and very crucial component where the user initiates communication with the chatbot. It is a visual element that allows users to interact with the chatbot and provides a means for the chatbot to communicate with the user. During the development of the chatbot, a key priority was placed on utilizing publicly available technologies and platforms. Specifically, the user interface of the chatbot was constructed using the Facebook Messenger API, which offers a range of tools and APIs designed to facilitate the creation of chatbots for the Facebook Messenger application. Through the use of the Facebook Messenger API, developers are able to develop customized chatbots that operate within the Facebook Messenger environment, allowing users to initiate conversations and engage with the chatbot.

Using the Facebook Messenger API enables the creation of chatbots that interact with users by sending and receiving messages, using templates and quick replies, and creating persistent menus. It also allows for receiving real-time updates about events that occur within the Facebook Messenger app, such as message delivery and receipt via webhooks. The Facebook Messenger API is accessible through RESTful interface, making it easy for developers to integrate with their existing systems and workflows. The API is designed to be scalable and flexible, allowing developers to build complex chatbots and conversational experiences with ease. A chatbot's UI components are an integral part of its overall design and functionality and should be carefully considered in order to provide a user-friendly and effective experience. Statistics show that social media users and mobile users heavily use Facebook Messenger. Moreover, the Facebook Messenger platform proved to be a powerful, sophisticated, and user-friendly platform to interact with the chatbot [46].

In January 2023, Facebook boasted a staggering 2.963 billion monthly active users, positioning it as the most active social media platform worldwide. Furthermore, Facebook Messenger, which remains inaccessible in China, amassed at least 931.0 million users globally in the same month. This impressive figure translates to a 14.9% penetration rate among individuals aged 13 and above who currently use Facebook Messenger, securing its place as the seventh most active social media platform globally [47,48].

The development of the chatbot involved the use of several Facebook Messenger APIs to facilitate user interactions. The Facebook Messenger messaging API was utilized to handle the exchange of messages between the user and the chatbot. This API, provided by the Facebook Messenger Platform, enables the chatbot to send messages to users through Facebook Messenger and receive responses in return.

To enhance the messaging experience and provide users with a more interactive experience, the Button Template API has been integrated with chatbot responses to user queries. This template has been used to provide the message recipient with a list of related questions to the query submitted by the user. We call this list Related Questions, which have been generated by the knowledge retrieval component and the user can post back those questions simply by clicking the button referring to the desired question.

In addition, the Customer Feedback Template has been used to provide users with the option of rating the chatbot response. This template has been associated with each chatbot response, which we call Rate Answer section. The Customer Satisfaction Score (CSAT) was also used, which allows users to rate the chatbot response from 1 to 5 stars.

### 3.2. Information Retrieval

In a chatbot architecture, the information retrieval (IR) component searches the chatbot's knowledge base for the most relevant data to address the user's query. It is a crucial part of a question-answering (QA) chatbot, as it fundamentally governs the quality of responses proffered by the chatbot. The chatbot employs two approaches for information retrieval: keyword matching and a primary approach called the Internet Wizard, which leverages the vast repository of the world wide web to obtain pertinent information.

#### 3.2.1. Keyword Matching

This is the first simple and straightforward approach used in information retrieval of the developed open-domain QA chatbot. The basic idea behind keyword matching is to compare the user's input with a predefined list of keywords or phrases to determine the most relevant response. For the keyword matching approach, the Wit.ai NLP model, was applied to detect keywords in user queries and MongoDB was used to store the keywords and corresponding answers in a structured database as the knowledgebase. The chatbot's knowledge base, in this case MongoDB, is pre-populated with a set of answers, and each answer is associated with a set of keywords that describe its content. When the user asks a question, the chatbot performs a keyword search through the wit.ai API to find the most relevant information in its knowledge base.

The keyword matching approach is used in such cases when the answer to a specific keyword or list of keywords needs to be hardcoded in the chatbot knowledge base for example questions related to the personality of the chatbot such as "What is your name?" otherwise the second approach which is Internet Wizard is used to answer user queries.

To facilitate the addition of keywords and answers to the MongoDB and Wit.ai databases, a user-friendly and sophisticated online platform has been developed. This platform streamlines the process of updating the knowledge base by eliminating the need for coding skills, making it easy and straightforward to maintain the database over time.

#### 3.2.2. Internet Wizard

Internet Wizard is the dominant approach of the IR component of the developed open-domain QA chatbot and the knowledgebase of the chatbot heavily depends on this approach. It is called Internet Wizard because it uses the world wide web knowledge to retrieve appropriate answer for the user quires. The world wide web (WWW) is a vast and constantly growing network of information, making it a vital resource for an open-domain chatbot knowledgebase. This approach combines several technologies, including data mining and machine learning, to find related documents on the web, index them, and process the content to locate knowledge that can be used for answering user inquiries.

The first step in this approach is searching the internet to find the most related documents to the user's queries. As a matter of fact, one of the most effective ways to access this information is through the use of search engines. Search engines are a fundamental component of the world wide web and provide users with an efficient means of accessing information. They have become the primary source of information for many individuals, who turn to them first when seeking information. Search engines are complex systems that use an interface to process and arrange documents based on their relevance to a given query [49]. In light of this fact, the approach used the most popular search engine worldwide, known as Google, and utilized its features. Google's popularity and effectiveness make it a go-to tool for information retrieval, as it uses a proprietary algorithm known as PageRank to rank web pages based on their relevance to a user's query. PageRank considers the link structure of the web and assesses an individual page's value based on it, facilitating efficient and expedient information retrieval [50]. Additionally, the vast size of Google's index, which was estimated to be over 55 billion individual web pages as of 2022 [51], makes it a powerful tool for discovering information on a wide range of topics.

As part of the information retrieval (IR) component, a web scraping technique is implemented to interact with Google search results and extract pertinent content from

featured snippets, the Google knowledge graph, and organic search results. This approach allows for the retrieval of relevant information directly associated with user queries.

The Internet Wizard mechanism relies on Google-featured snippets as the primary source of information for answering user queries. Featured snippets are the latest enhancement to Search Engine Results Pages (SERPs), which display relevant information from web pages along with their source URL above organic search results. This provides users with quick and accurate answers to their queries, improving their search experience and efficiency [52]. To generate its featured snippets, Google uses a complex set of algorithms. The exact algorithms used by Google are not publicly disclosed. However, studies show several factors that influence the generation of featured snippets including the relevance and quality of the content, the source of the content, rank of the page, the user's search query, the page structure and formatting, multiple keyword inclusion in the website's content, and the use of different keyword locations, such as headings, titles, URLs, paragraphs, image ALT, link, and the frequency of the query [53,54].

The Internet Wizard leverages the Google knowledge graph as the second source of knowledge for responding to user queries. The Google knowledge graph is a semantic search engine that aims to provide more relevant and comprehensive information on entities, including people, places, things, and events. This is achieved by integrating structured data from multiple sources and processing it using machine learning algorithms to identify new relationships. By representing information using entities and their relationships, the knowledge graph provides users with constantly evolving and updating information. However, maintaining the vast amount of data and ensuring its completeness and accuracy presents significant challenges [55].

During the next phase of Internet Wizard, if both the Google featured snippet and Google knowledge graph fail to provide appropriate answers to user queries, the next step utilizes Google organic search results. Google organic search results are the web pages that appear in response to a user's search query and are ranked based on relevance and popularity. These results are generated by Google's algorithms, which use complex mathematical formulas to determine the relevance and popularity of web pages. The algorithms take into account a variety of factors, including the content of the web page, the number of links pointing to the page, the relevance of the page to the search query, and the overall quality of the website. Google's organic search results are powered by the PageRank and RankBrain algorithms. PageRank analyzes the number and quality of links pointing to a web page and assigns a relevance score, while RankBrain uses machine learning to provide more relevant and accurate results for complex and conversational searches [56,57].

The mechanism takes the first five organic search results pages and feeds their content to the TF-IDF algorithm after removing HTML tags. TF-IDF is a statistical method used in information retrieval and natural language processing to determine the importance of a word in a document or corpus. It calculates word relevance based on its frequency in a document (term frequency) and rarity in the entire corpus (inverse document frequency), resulting in a score that reflects both. TF-IDF is widely used in information retrieval, text classification, and text clustering to assess document relevance [58,59]. The TF-IDF algorithm selects the most relevant candidate document from the top five ranked pages, based on the user's query, to generate a response to the query in the next step.

The candidate document and user query are inputted into a TensorFlow Question and Answer (QA) model after being retrieved by TF-IDF. This model uses a pre-trained BERT model fine-tuned on the SQuAD 2.0 dataset to answer user questions based on the candidate document's content. TensorFlow is an open-source library developed by Google Brain [60], and has emerged as a powerful platform for QA tasks due to its scalability and flexibility. A widely used approach for QA tasks is to use pre-trained transformer models, such as BERT [28], to encode input text and questions into a fixed-length representation and use a classifier to predict the answer span. TensorFlow provides pre-built models and libraries for implementing BERT and other transformer models, making it easy to fine-tune

these models for QA tasks [61]. The use of pre-trained models reduces the amount of labeled data needed for training and improves performance by leveraging a large amount of general-purpose language information learned during pre-training.

In the event that the TensorFlow question–answer model is unable to provide an answer based on the content of the candidate document, then the link to the candidate document with the default text “the following link provides a detailed answer” is provided as the answer to the user’s query. The user can click the link to open the document inside the chat window without opening an external browser app. This document usually contains the most relevant information and explanation for the user’s query.

### 3.3. NLP

Chatbot applications require natural language processing (NLP) as an essential component to understand and interpret the user’s language. NLP techniques include text classification, named entity recognition, sentiment analysis, and language translation, which are performed using rule-based methods, machine learning algorithms, and deep learning techniques [62]. For the developed chatbot, the wit.ai NLP platform was employed to handle the first part of the NLP component. Wit.ai is a popular NLP platform that makes it easy for developers to incorporate natural language processing into their applications, particularly chatbots which use a combination of rule-based and machine learning-based techniques to process natural language input and extract information such as entities and intents [63]. This information can then be used to trigger actions within an application or generate a response. The platform also enables developers to train customized models that can recognize specific entities, intents, and actions within a particular domain. As the expected user query structure was similar to our previous research, we used the same wit.ai application previously created and trained for that study.

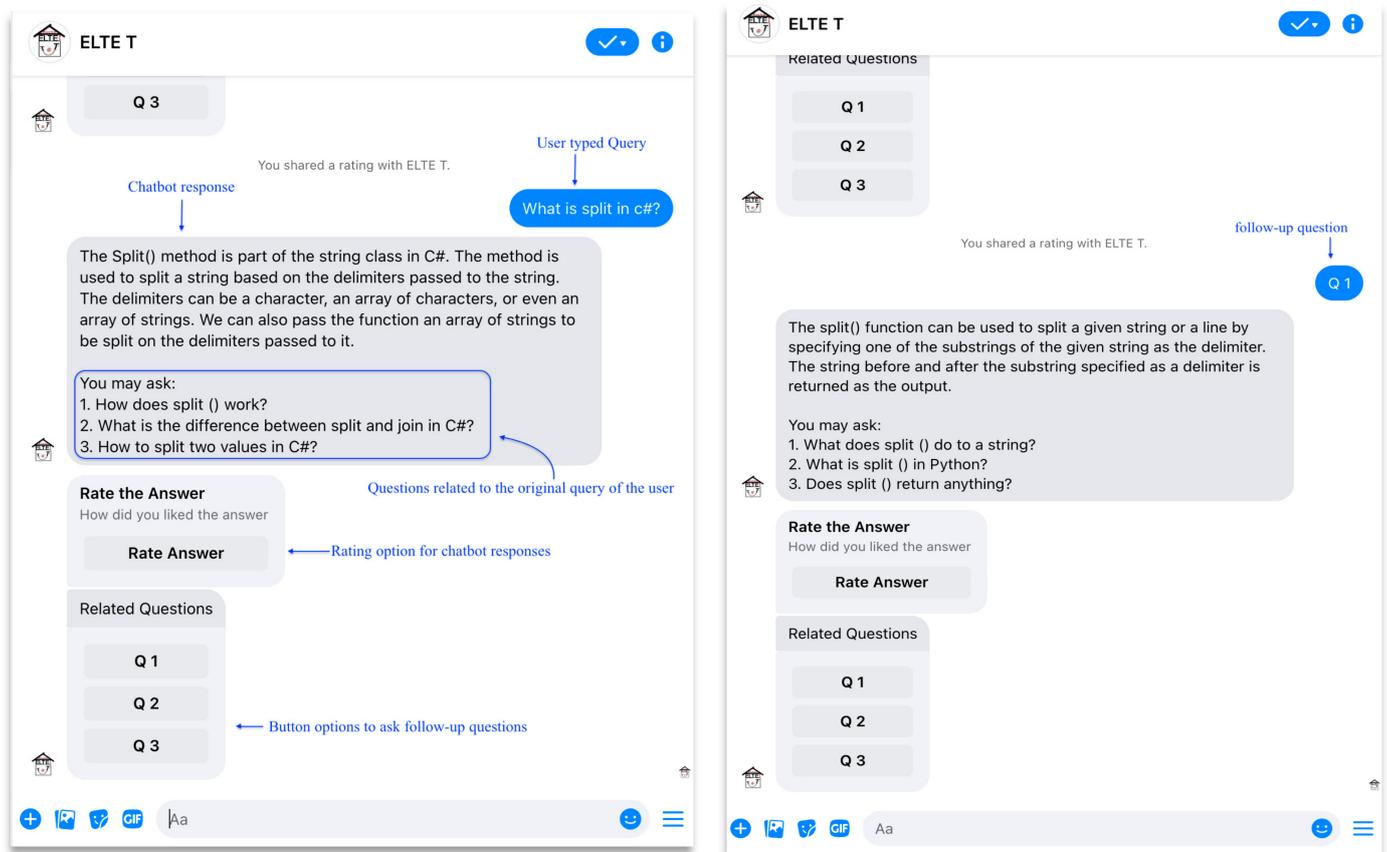
The wit.ai model was trained to detect entities, or more specifically keywords, in each user query. Later, these entities were used to find the answer in the database.

The Tensorflow Question and Answer model, based on the BERT model, was also used in the chatbot architecture to provide answers to user queries. This model is created based on the Bidirectional Encoder Representations from Transformers (BERT) model and fine-tuned on SQuAD 2.0 dataset. BERT, developed by Google, is a transformer-based model that has been fine-tuned for a wide range of NLP tasks and has set new state-of-the-art performance on several benchmark datasets [28].

### 3.4. Response Generation Component

This component is responsible for generating responses to user input and is a critical element in determining the effectiveness of the chatbot. This component is tightly integrated with the information retrieval component. The response-generating component uses Facebook messenger APIs more specifically the pages\_messaging API to send the response generated from the IR component. The Response Generation Component integrates two methods from the IR component: the Retrieval-based model using keyword matching to select a suitable response from a database, and the Generative model called Internet Wizard, which leverages Google snippets, knowledge graphs, TF-ID algorithm, and Tensorflow Question and Answer model to generate responses.

In addition, the Response Generation Component adds a list of the three related questions to the response generated by the IR component for each chatbot response. As showcased in Figure 2, associated questions are the most related questions that people have asked about the same topic or keyword on Google search engine. The Messaging API’s button template is used to create a button for each related question, enabling users to easily ask follow-up questions by selecting the appropriate button.



**Figure 2.** Examples of chatbot responses to user queries.

It is hypothesized that this feature of the chatbot will enable users to dig deeper into the query in mind and become more informed about related topics, which likely leads to more engagement between the chatbot and the user. Furthermore, the Customer Feedback Template Messaging API is used in this component to gather feedback from users in real time. This template is associated with each chatbot response and is known as the “Rate the Answer” section. When the user clicks the “Rate the Answer” button, a rating window appears that allows users to rate their satisfaction with the chatbot response on a scale of 1 to 5 using the Customer Satisfaction Score (CSAT) metric. CSAT is a commonly used metric for measuring user satisfaction through surveys, providing businesses with a quick and easy way to monitor customer satisfaction over time [64]. The ratings collected are saved in the MongoDB database for analysis of chatbot response quality and user satisfaction.

#### 4. Pilot Study

To evaluate the performance of the implemented mechanism and the open-domain question QA chatbot features, a pilot study was conducted. Participants from the first-year programming course at Eotvos Lorand University were enlisted, with encouragement to utilize the chatbot for any questions they had regarding the programming course, other courses, or unrelated topics. This course covers the basics of algorithms and programming and uses C# as the programming language. The study featured 35 participants of different nationalities, including 13 females and 22 males. It was carried out between 1 December 2022 and 15 January 2023.

#### 4.1. Data Collection

The data generated from the QA chatbot and user interaction were stored in two MongoDB collections (data and feedbacks) on a virtual node provided by ELTE where the chatbot code was deployed. MongoDB, a document-oriented NoSQL database system, stored the data in collections of JSON-like documents in BSON format.

The collected data involved different metrics, such as the user's query and the origin, which could either be entered by the user at the chatbot prompt or derived from the Related Questions feature of the chatbot. This metric is used for checking the effectiveness of the Related Questions feature of the chatbot in motivating the user to explore and discover more knowledge about the topic. The data further encompassed a distinct variable used to distinguish whether the answer obtained from the world wide web was directly from Google features, such as Google featured snippets or Google knowledge graph, or from a Tensorflow QA model dependent on the candidate document selected by TF-IDF. Furthermore, a variable was assigned to keep track of all the web documents used by the mechanism to generate the answer. Finally, another variable was created to capture the user's rating and satisfaction with the chatbot's response.

#### 4.2. Evaluation Method

Chatbot technology has undergone continuous evolution, and it is expected to continue improving in the foreseeable future. Despite their widespread use, the standard for evaluating chatbots has not been firmly established and may become obsolete as technology advances. Even so, the evaluation of chatbots is crucial to their development and success. In this research, the user satisfaction evaluation method has been used to evaluate the mechanism and features of the developed chatbot.

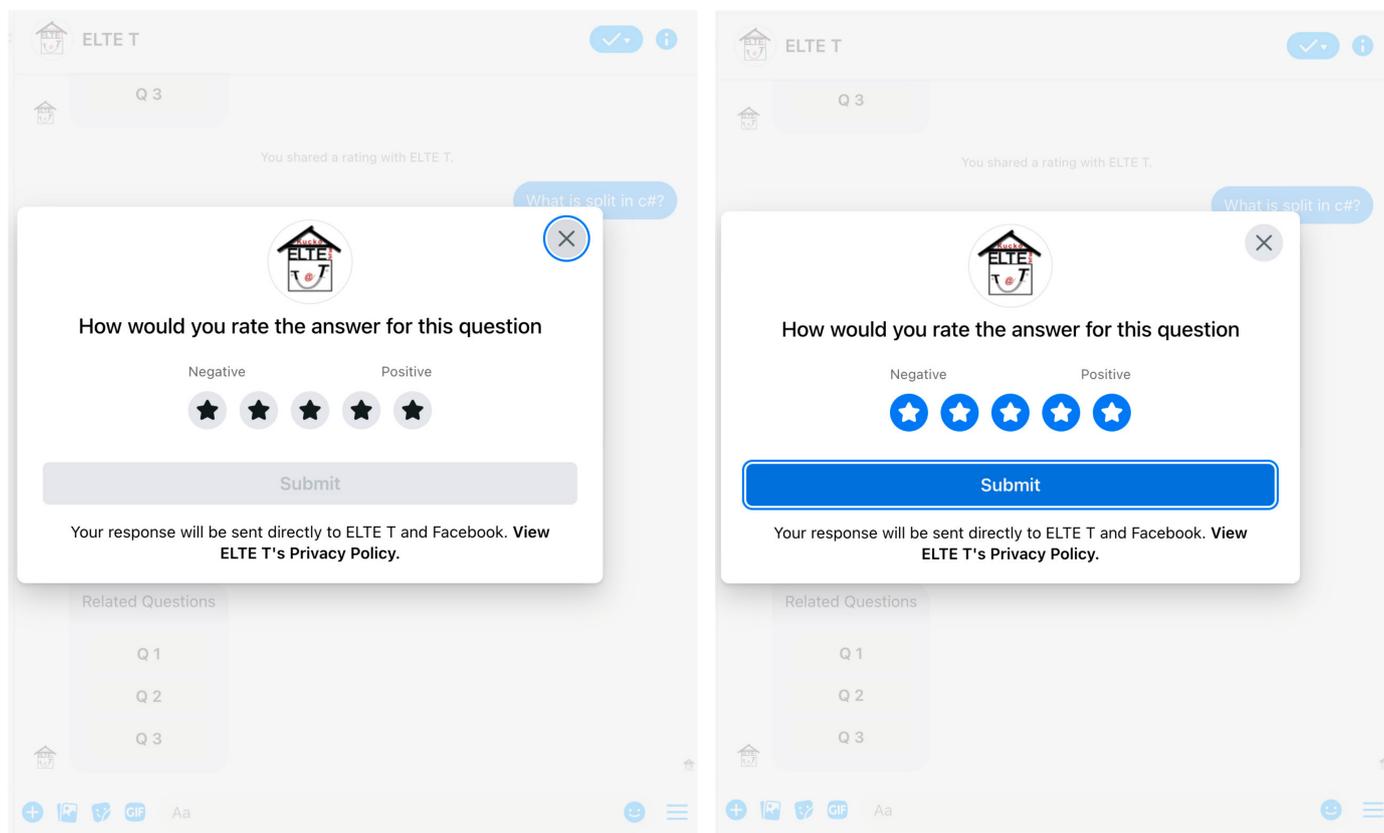
User satisfaction is a widely adopted evaluation method in chatbot assessment, where users interact with the chatbot and rate their satisfaction using a Likert scale or a similar measure [65]. User satisfaction can be measured at two levels: session and turn level. At the session level, user satisfaction is measured based on the overall experience of the entire interaction between the user and the chatbot in one session. Turn level user satisfaction is a metric that reflects a user's satisfaction with each turn in the conversation. The turn level is defined as the interaction between a user and a chatbot, starting with a user message and ending in a chatbot response.

This research uses turn-level evaluation to enable users to evaluate each chatbot's response. This approach is a commonly used approach for assessing user satisfaction with open-domain QA chatbots, which allows users to express their preferences and opinions for each interaction. This method is preferred due to the diverse nature of topics that users may ask about, which requires a more flexible and independent evaluation approach. Studies have shown that incorporating user feedback through turn-level evaluations can improve chatbot performance and user engagement over time. The effectiveness and reliability of the turn-level evaluation method for chatbot evaluation has been documented in several studies [39,66,67].

The turn-level user satisfaction evaluation method in this research is implemented using the Customer Feedback Template Messaging API. In this context, a template known as the "Rate the Answer" section has been incorporated into each chatbot response. This template prompts the user to rate their satisfaction with the chatbot response by presenting a rating window when the user clicks the "Rate the Answer" button. The rating scale ranges from 1 to 5 and is based on the Customer Satisfaction Score (CSAT) methodology as illustrated in Figure 3.

The CSAT methodology is a widely adopted approach within the customer service industry to gauge customer satisfaction levels with a given product or service. This approach involves administering surveys to obtain a rating score, which can be numerical or symbolic (e.g., stars or smiley faces). The score ranges from 1 to 5, where higher values indicate higher satisfaction. The CSAT methodology has found applicability within the field of human-computer interaction (HCI) in assessing the usability and overall user

experience of software applications including chatbots and conversational agents. CSAT scores serve as a valuable tool in assessing user interface effectiveness and holistic user experience [68].



**Figure 3.** Evaluating chatbot responses with CSAT.

## 5. Data Analysis and Results

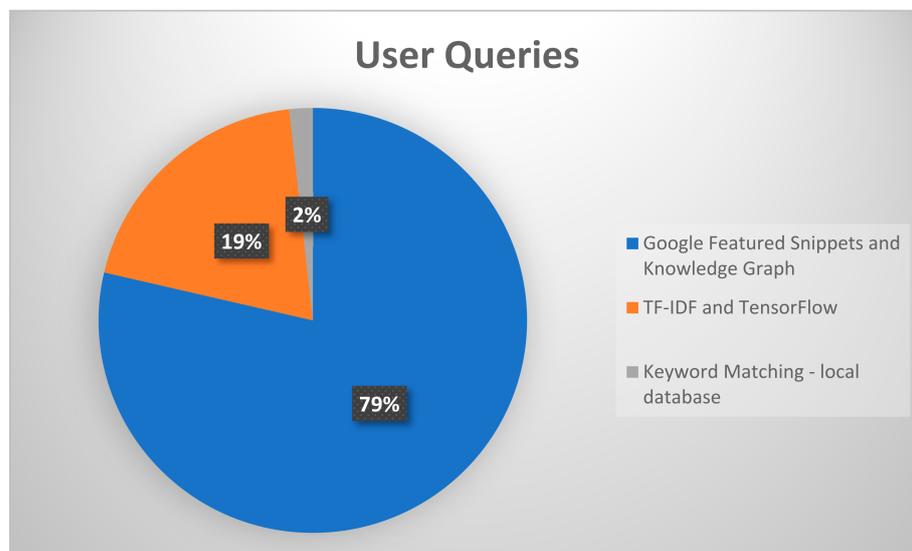
### 5.1. Analysis of a Chatbot's User Queries

The chatbot received a total of 1132 queries from 35 users. Out of these, 20 queries were about the chatbot's personality, such as "Who are you?" and "Where are you from?" The chatbot's local knowledge base had hardcoded answers for these queries, such as "I am designed and developed by the Kuckó lab team at Eötvös Loránd University. Ask questions about other topics! Such as: What is a higher-order function in functional programming?".

The other 1112 questions were diverse and covered a wide range of topics. Popular subjects included programming and coding concepts, mathematics, science and technology, health, and sports. Other miscellaneous topics were also asked about. It is noteworthy that while a significant number of inquiries received were related to course content, particularly programming and mathematics, many students posed questions regarding a diverse range of topics. For instance, one student asked about capitalism and the 1776 revolution, while another inquired, "Is there any pharmacy in elte lagymanyos campus?" and reached out for help with depression, writing "I am depressed can you help me?". Other diverse questions such as "How is Ukrainian beetroot soup called? When is the final match of World Cup 2022? Is VW atlas bigger than Telluride?" were also asked. These queries indicate that the chatbot was being used for a variety of purposes beyond course content, highlighting the need for chatbots to be capable of addressing a wide range of topics.

To address these queries, the chatbot utilized the Internet Wizard mechanism from the world wide web. Out of the 1112 queries, 890 answers were generated using Google feature snippets and the Google knowledge graph, while the remaining 222 responses were

produced with the aid of the TF-IDF and Tensorflow modules. Figure 4 presents a graphical representation of this information.



**Figure 4.** Answer source for user queries.

### 5.2. Analysis of Web Documents Used by Internet Wizard Mechanism

The Internet Wizard sub-component of the chatbot uses 763 unique web documents to generate answers. Each of these documents was employed at least once to generate answers to user queries. Analysis of resource metrics revealed that Wikipedia was the most frequently used web source with a record count of 116. This suggests that users who accessed the chatbot were using it as a source of general knowledge and information. This is followed by the URL <https://www.simplilearn.com/tutorials/c-tutorial/>, which has a record count of 27, which indicates that users were interested in programming and mathematical topics. It is noteworthy that the URLs with the highest record counts are predominantly educational resources. For instance, <https://www.javatpoint.com/pyhton-factorial-number> and <https://www.programiz.com/c-programming>, which have eight and seven record counts respectively. This suggests that users were inquiring about information on how to solve programming and math problems related to their courses. The other websites on the list are more specialized and focus on particular topics such as information technology, science, sports, health, and many other diverse topics. These entries suggest that users use the chatbot to answer questions about a diverse range of interests beyond just programming and math. In Figure 5, the top twenty most frequently used web documents by the chatbot to generate answers to user queries were identified and classified into separate categories in the pie chart. The remaining web documents, which were referenced less frequently and therefore demonstrated a high degree of diversity, have been aggregated into a collective category labeled “others”. This category represents web documents that were used between one and three times, consisting of 743 unique web documents. This finding indicates that the Internet Wizard mechanism utilized a wide range of web documents to generate responses to user queries, which can be attributed to the diversity of the queries’ topics.

### 5.3. Distribution of Queries in the Chatbot

The distribution of queries in Figure 6 indicates that the Related Questions feature was the predominant source, accounting for 55.6% of the records, compared with user-input questions, which accounted for 44.4% of the records. This implies that a substantial proportion of the questions asked were generated through the Related Questions feature, indicating its ability to capture users’ attention and stimulate their interest in seeking further knowledge about the original query.



Analysis of the feedback collection documents presented in Table 1 suggests that the majority of chatbot responses were sourced from Google featured snippets and knowledge graphs. Specifically, for questions generated by the Related Questions feature, 453 answers were provided from this source, compared with 238 answers provided for questions inputted by users.

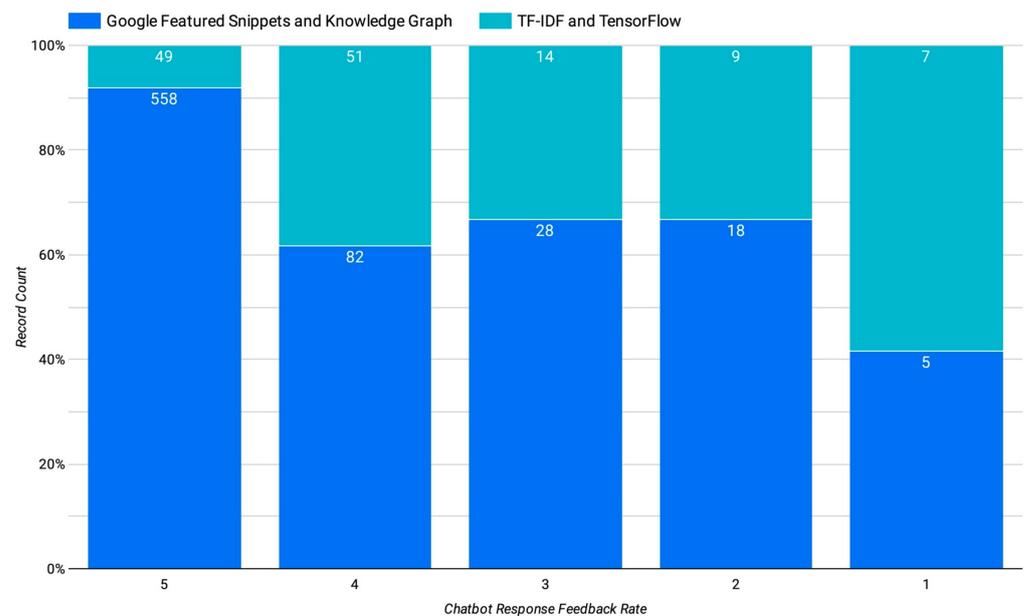
**Table 1.** Average feedback rate for chatbot responses by query and answer source.

Source of Query	Source of Answer	Record Count	Response Feedback Rate Average
Related Questions Feature	Google Featured Snippets and Knowledge Graph	453	4.69
User-input Questions	Google Featured Snippets and Knowledge Graph	238	4.7
User-input Questions	TF-IDF and TensorFlow	72	3.81
Related Questions Feature	TF-IDF and TensorFlow	58	4.17

Similarly, for questions input by users, most answers were sourced from Google featured snippets and knowledge graphs, with only 72 coming from TF-IDF and TensorFlow. This finding indicates that Google’s featured snippets and knowledge graph sections of the Internet Wizard mechanism played a significant role in providing answers.

The feedback ratings for the answers were generally high, with an average rating of 4.69 for questions sourced from the Related Questions feature and Google featured snippets and knowledge graphs, and an average rating of 4.70 for questions inputted by users and sourced from the same source.

Figure 7 depicts that the majority of responses sourced from Google featured snippets and knowledge graphs received a rating of 5. However, when TF-IDF and TensorFlow were used as sources of answers for user-input questions, the average feedback rating decreased to 3.81. Responses sourced from TF-IDF and TensorFlow received lower ratings compared with those sourced from Google’s featured snippets and knowledge graphs, as illustrated in Figure 7, indicating a lower level of satisfaction with the answers provided by these sources.



**Figure 7.** User’s feedback rating for chatbot responses.

Regarding the number of records, the highest count of 453 was for answers sourced from Google featured snippets and knowledge graphs for questions generated by the Related Questions feature. The lowest count of 58 was for answers sourced from TF-IDF

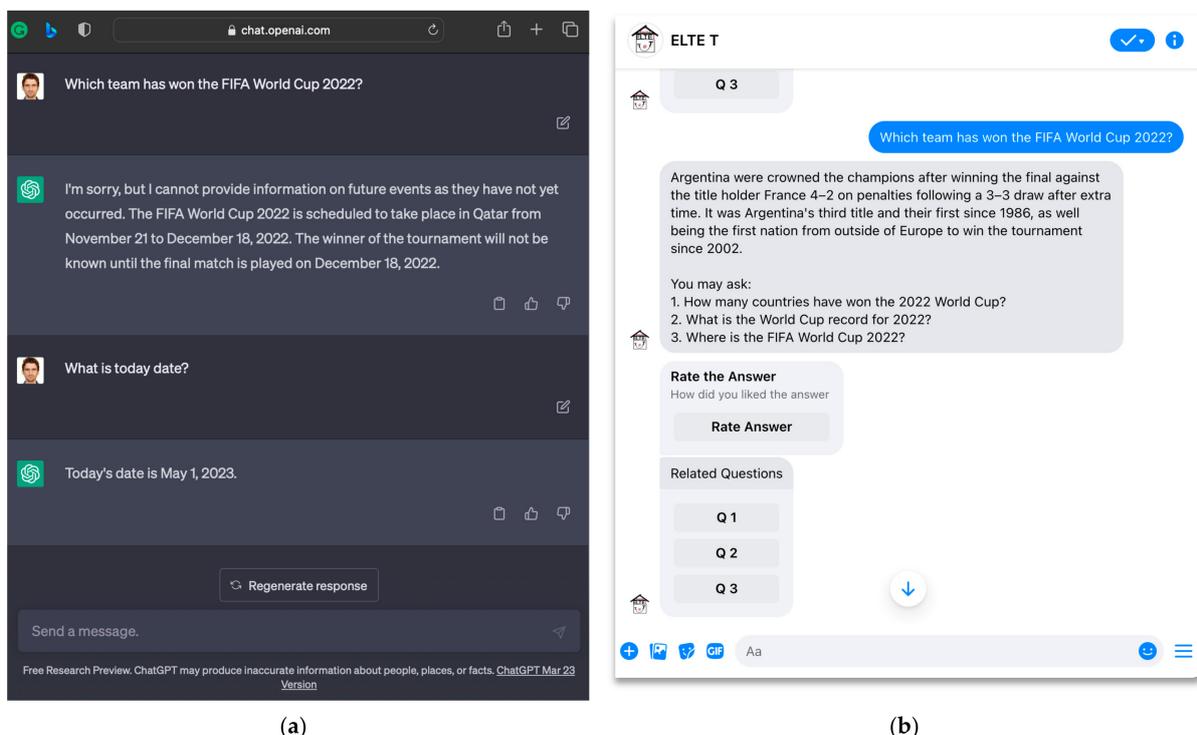
and TensorFlow for questions generated by the Related Questions feature. The average feedback rate was 4.34, indicating that the majority of users were highly satisfied with the chatbot's responses. A feedback rating of 4 or above is generally considered positive, indicative of the chatbot's ability to accurately answer a wide range of questions and provide users with the information they need satisfactorily.

Overall, the results suggest that Google featured snippets and knowledge graphs were the primary sources for generating answers in the Internet Wizard mechanism. The high feedback ratings for answers sourced from this section Internet Wizard mechanism indicate that users were generally satisfied with the answers provided. Based on the feedback analysis, the Internet Wizard mechanism appears to be generally effective at answering user queries with a high level of user satisfaction. However, there is potential for improvement, particularly with regard to the use of TF-IDF and TensorFlow as sources of generating answers.

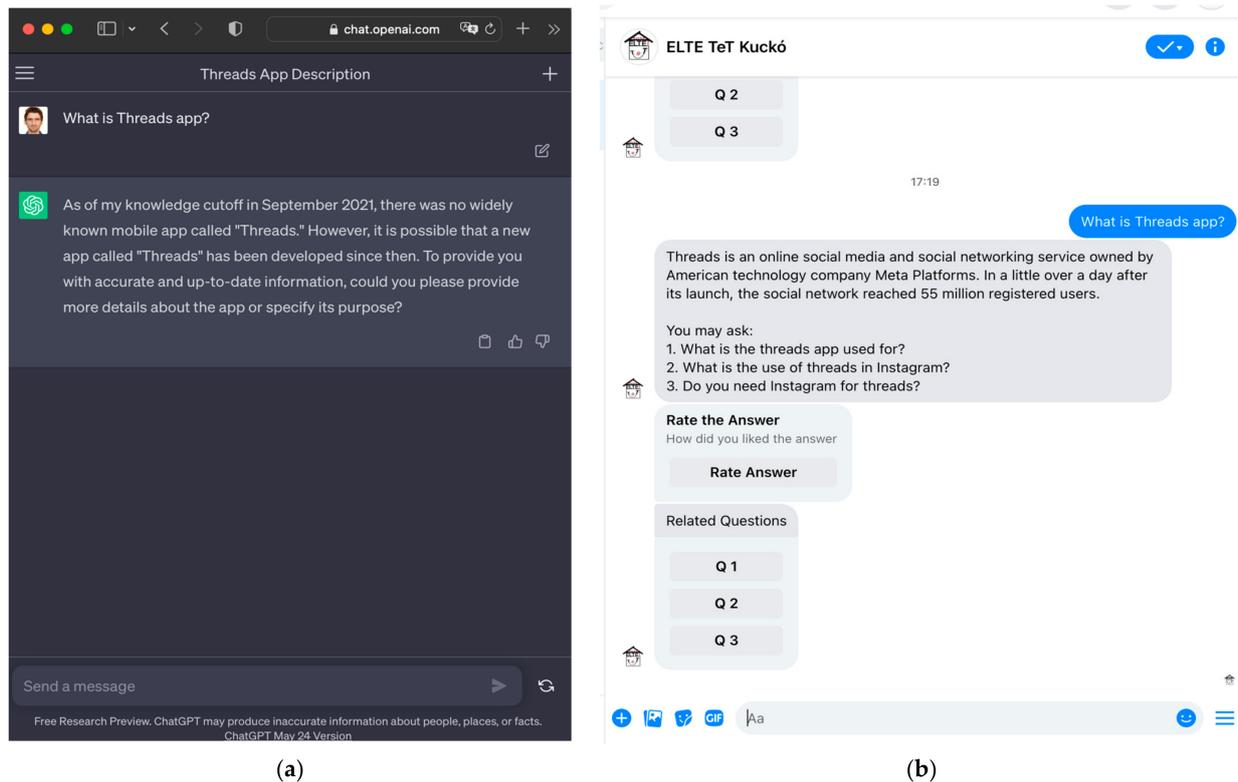
### 5.5. Comparison to Static Language Model

The Internet Wizard extracts information from the vast world wide web and continually updates its knowledge base in real time without requiring any fees. This cost-effective approach enables the chatbot to provide flexible and adaptable responses to queries that it has not been explicitly trained on, thus providing a more flexible and adaptable solution. Unlike statistical knowledgebases, the dynamic knowledge retrieval of the Internet Wizard provides a diverse and up-to-date knowledge acquisition.

Statistical language models, including large ones such as GPT, are limited by their training data and have a static and time-limited knowledge base. This means that they are unable to respond to queries based on information acquired after a specific time period, such as 2021. For instance, Figures 8a and 9a demonstrate that ChatGPT struggles to answer questions about events and products that occurred in 2022 and 2023. In contrast, Figures 8b and 9b show how the Internet Wizard mechanism effectively addresses such queries by continuously retrieving the most recent and relevant information from the world wide web.



**Figure 8.** (a) Temporal knowledge base of the GPT-3.5 language model; (b) real-time knowledge base of the Internet Wizard mechanism.



**Figure 9.** (a) Temporal knowledge base of the GPT-3.5 language model; (b) real-time knowledge base of the Internet Wizard mechanism.

The integration of the Internet Wizard mechanism with GPT can serve as a solution to the limitations of a static knowledge base in language models. By continuously retrieving the most recent and relevant information available on the web, the Internet Wizard mechanism enables the model to expand its knowledge base, resulting in more comprehensive and up-to-date responses. This combination of dynamic knowledge retrieval and advanced language modeling can result in a chatbot that provides accurate and informative responses to a wide range of user queries in a human-like style.

## 6. Conclusions, Limitations, and Future Works

### 6.1. Conclusions

This study assessed the effectiveness of the Internet Wizard mechanism for improving the knowledge base of an open-domain QA chatbot. By employing search engine features, data mining techniques, and a machine learning module, the Internet Wizard mechanism dynamically retrieved related knowledge from the web, enabling the chatbot to generate responses to a wide range of topics.

The study demonstrated that the use of Google featured snippets and knowledge graphs played a crucial role in providing accurate responses to user queries, resulting in high levels of user satisfaction with the generated answers. Moreover, the chatbot and the Internet Wizard mechanism handled both educational and non-educational topics satisfactorily, as evidenced by the high rate of feedback and user satisfaction.

The use of dynamic knowledge-retrieving approaches, such as the Internet Wizard mechanism, offers a powerful tool for generating answers to a wide range of questions. This approach can complement the strengths of statistical language models, providing a comprehensive and flexible solution for natural language processing tasks. The success of the chatbot in addressing a broad range of topics underscores the potential of such mechanisms in developing a dynamic knowledge base for an open-domain QA chatbot.

In addition, the study highlights the importance of using the Related Questions feature, as it contributed to the majority of the questions asked by users indicating that users were more inclined to leverage the chatbot as an information source and seek follow-up questions. The findings suggest that a chatbot that can handle a wide variety of topics, leverages the Related Questions feature, and provides accurate answers, can be a valuable tool for students seeking information beyond their coursework.

In conclusion, the utilization of the Internet Wizard mechanism, combined with the Related Questions feature, offers a promising approach for developing a dynamic knowledge base for an open-domain QA chatbot that can handle a wide range of topics. The success of the chatbot in addressing a broad range of questions, along with the high rate of feedback and user satisfaction, highlights the potential of this approach for providing accurate and informative responses to users seeking information.

### 6.2. Limitations

The present study demonstrates the efficacy of the Internet Wizard mechanism in augmenting the chatbot's proficiency in handling diverse topics. However, inconsistent performance was observed in the sections that employed TF-IDF and TensorFlow.

Specifically, the TensorFlow model intermittently failed to predict accurate responses based on the candidate document generated by TF-IDF.

This limitation may be attributed to either a paucity of pertinent information in the document content or the requirement for further refinement of the web document context through text fine-tuning techniques. To address these limitations and enhance the performance of the Internet Wizard mechanism, several potential solutions can be considered. One approach involves retrieving pertinent knowledge for the user's query from multiple documents and synthesizing them into a comprehensive knowledge base that can be fed into the TensorFlow model. Additionally, fine-tuning the TensorFlow model itself for specific tasks with relevant training data can potentially improve its precision and effectiveness.

### 6.3. Future Works

In terms of future research directions, there exists an opportunity to investigate the potential benefits of integrating the Internet Wizard mechanism with advanced natural language processing techniques, such as neural and Large Language Models (LLMs). This can serve to further enhance the chatbot's overall performance, thereby offering a more human-like chatting experience to the user, while also providing access to a dynamic and up-to-date knowledge base. Additionally, the potential solutions that have been suggested for addressing the limitations of the Internet Wizard mechanism can be explored in further research.

**Author Contributions:** Conceptualization, K.M. and M.T.-S.; methodology, K.M. and M.T.-S.; software, K.M.; validation, K.M.; formal analysis, K.M.; investigation, K.M.; resources, M.T.-S.; data curation, K.M.; writing—original draft preparation, K.M.; writing—review and editing, K.M. and M.T.-S.; visualization, K.M.; supervision, M.T.-S.; project administration, M.T.-S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Grudin, J.; Jacques, R. Chatbots, Humbots, and the Quest for Artificial General Intelligence. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, UK, 4–9 May 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1–11.
2. Smutny, P.; Schreiberova, P. Chatbots for Learning: A Review of Educational Chatbots for the Facebook Messenger. *Comput. Educ.* **2020**, *151*, 103862. [\[CrossRef\]](#)
3. Okonkwo, C.W.; Ade-Ibijola, A. Python-Bot: A Chatbot for Teaching Python Programming. *Eng. Lett.* **2020**, *29*, 25–34.
4. Weizenbaum, J. ELIZA—A Computer Program for the Study of Natural Language Communication between Man and Machine. *Commun. ACM* **1966**, *9*, 36–45. [\[CrossRef\]](#)
5. Shawar, B.A.; Atwell, E. ALICE Chatbot: Trials and Outputs. *Comput. Syst.* **2015**, *19*, 625–632.
6. Razzaki, S.; Baker, A.; Perov, Y.; Middleton, K.; Baxter, J.; Mullarkey, D.; Sangar, D.; Taliercio, M.; Butt, M.; Majeed, A.; et al. A Comparative Study of Artificial Intelligence and Human Doctors for the Purpose of Triage and Diagnosis. *arXiv* **2018**, arXiv:180610698.
7. Feldman, M.J.; Hoffer, E.P.; Barnett, G.O.; Kim, R.J.; Famiglietti, K.T.; Chueh, H.C. Impact of a Computer-Based Diagnostic Decision Support Tool on the Differential Diagnoses of Medicine Residents. *J. Grad. Med. Educ.* **2012**, *4*, 227–231. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Wube, H.D.; Esubalew, S.Z.; Weldesellasie, F.F.; Debelee, T.G. Text-Based Chatbot in Financial Sector: A Systematic Literature Review. *Data Sci. Finance Econ.* **2022**, *2*, 232–259. [\[CrossRef\]](#)
9. Mogaji, E.; Balakrishnan, J.; Nwoba, A.C.; Nguyen, N.P. Emerging-Market Consumers' Interactions with Banking Chatbots. *Telemat. Inform.* **2021**, *65*, 101711. [\[CrossRef\]](#)
10. Følstad, A.; Taylor, C. Investigating the User Experience of Customer Service Chatbot Interaction: A Framework for Qualitative Analysis of Chatbot Dialogues. *Qual. User Exp.* **2021**, *6*, 6. [\[CrossRef\]](#)
11. Ngai, E.W.T.; Lee, M.C.M.; Luo, M.; Chan, P.S.L.; Liang, T. An Intelligent Knowledge-Based Chatbot for Customer Service. *Electron. Commer. Res. Appl.* **2021**, *50*, 101098. [\[CrossRef\]](#)
12. Clark, L.; Pantidi, N.; Cooney, O.; Doyle, P.; Garaialde, D.; Edwards, J.; Spillane, B.; Gilmartin, E.; Murad, C.; Munteanu, C.; et al. What Makes a Good Conversation? Challenges in Designing Truly Conversational Agents. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, UK, 4–9 May 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1–12.
13. Coronado, M.; Iglesias, C.A.; Carrera, Á.; Mardomingo, A. A Cognitive Assistant for Learning Java Featuring Social Dialogue. *Int. J. Hum.-Comput. Stud.* **2018**, *117*, 55–67. [\[CrossRef\]](#)
14. Daud, S.H.M.; Teo, N.H.I.; Zain, N.H.M. Ejava Chatbot for Learning Programming Language: Apost-Pandemic Alternative Virtual Tutor. *Int. J.* **2020**, *8*, 3290–3298.
15. Zawacki-Richter, O.; Marín, V.I.; Bond, M.; Gouverneur, F. Systematic Review of Research on Artificial Intelligence Applications in Higher Education—Where Are the Educators? *Int. J. Educ. Technol. High. Educ.* **2019**, *16*, 39. [\[CrossRef\]](#)
16. Cunningham-Nelson, S.; Boles, W.; Trouton, L.; Margerison, E. A Review of Chatbots in Education: Practical Steps Forward. In Proceedings of the 30th Annual Conference of Australasian Association for Engineering Education, Brisbane, Australia, 8–11 December 2019; Engineers Australia: Brisbane, Australia, 2019; pp. 299–306.
17. Ranoliya, B.R.; Raghuwanshi, N.; Singh, S. Chatbot for University Related FAQs. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 1525–1530.
18. Sinha, S.; Basak, S.; Dey, Y.; Mondal, A.K. An Educational Chatbot for Answering Queries. *Adv. Intell. Syst. Comput.* **2019**, *937*, 55–60.
19. Durall, E.; Kapros, E. Co-Design for a Competency Self-Assessment Chatbot and Survey in Science Education. In Proceedings of the Learning and Collaboration Technologies. Human and Technology Ecosystems: 7th International Conference, LCT 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Proceedings, Part II 22, Copenhagen, Denmark, 19–24 July 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 13–24.
20. Benotti, L.; Martnez, M.C.; Schapachnik, F. A Tool for Introducing Computer Science with Automatic Formative Assessment. *IEEE Trans. Learn. Technol.* **2017**, *11*, 179–192. [\[CrossRef\]](#)
21. Ismail, M.; Ade-Ibijola, A. Lecturer's Apprentice: A Chatbot for Assisting Novice Programmers. In Proceedings of the 2019 international multidisciplinary information technology and engineering conference (IMITEC), Vanderbijlpark, South Africa, 21–22 November 2019; pp. 1–8.
22. Dsouza, R.; Sahu, S.; Patil, R.; Kalbande, D.R. Chat with Bots Intelligently: A Critical Review & Analysis. In Proceedings of the 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), IEEE, Mumbai, India, 20–21 December 2019; pp. 1–6.
23. Chinedu, O.; Ade-Ibijola, A. Chatbots Applications in Education: A Systematic Review. *Comput. Educ. Artif. Intell.* **2021**, *2*, 100033. [\[CrossRef\]](#)
24. Ondáš, S.; Pleva, M.; Hládek, D. How Chatbots Can Be Involved in the Education Process. In Proceedings of the 2019 17th International Conference on Emerging Elearning Technologies and Applications (ICETA), IEEE, Starý Smokovec, Slovakia, 21–22 November 2019; pp. 575–580.

25. Hone, K.; El Said, G. Exploring the Factors Affecting MOOC Retention: A Survey Study. *Comput. Educ.* **2016**, *98*, 157–168. [CrossRef]
26. Eom, S.; Wen, J.; Ashill, N. The Determinants of Students' Perceived Learning Outcomes and Satisfaction in University Online Education: An Empirical Investigation. *Decis. Sci. J. Innov. Educ.* **2006**, *4*, 215–235. [CrossRef]
27. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach 2019. *arXiv* **2019**, arXiv:1907.11692.
28. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
29. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models Are Few-Shot Learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; Curran Associates Inc.: Red Hook, NY, USA, 2020.
30. Baumgartner, J.; Zannettou, S.; Keegan, B.; Squire, M.; Blackburn, J. The Pushshift Reddit Dataset 2020. In Proceedings of the Fourteenth International AAAI Conference on Web and Social Media, Atlanta, GA, USA, 8–11 June 2020.
31. Rashkin, H.; Smith, E.M.; Li, M.; Boureau, Y.-L. Towards Empathetic Open-Domain Conversation Models: A New Benchmark and Dataset. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 5370–5381.
32. Adiwardana, D.; Luong, M.-T.; So, D.R.; Hall, J.; Fiedel, N.; Thoppilan, R.; Yang, Z.; Kulshreshtha, A.; Nemade, G.; Lu, Y.; et al. Towards a Human-like Open-Domain Chatbot. *arXiv* **2020**, arXiv:2001.09977.
33. Roller, S.; Dinan, E.; Goyal, N.; Ju, D.; Williamson, M.; Liu, Y.; Xu, J.; Ott, M.; Shuster, K.; Smith, E.M.; et al. Recipes for Building an Open-Domain Chatbot. *arXiv* **2020**, arXiv:2004.13637.
34. Krassmann, A.; Flach, J.; Raquel, A.; Tarouco, L.; Bercht, M. A Process for Extracting Knowledge Base for Chatbots from Text Corpora. In Proceedings of the 2019 IEEE Global Engineering Education Conference (EDUCON), Dubai, United Arab Emirates, 8–11 April 2019; pp. 322–329.
35. Nishimura, R.; Watanabe, Y.; Nishimura, R.; Okada, Y. Confirmed Knowledge Acquisition Using Mails Posted to a Mailing List. In Proceedings of the Natural Language Processing—IJCNLP 2005, Jeju Island, Republic of Korea, 11–13 October 2005; Dale, R., Wong, K.-F., Su, J., Kwong, O.Y., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 131–142.
36. Shrestha, L.; McKeown, K. Detection of Question-Answer Pairs in Email Conversations. In Proceedings of the COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics, Geneva, Switzerland, 23–27 August 2004; pp. 889–895.
37. Huang, J.; Zhou, M.; Yang, D. Extracting Chatbot Knowledge from Online Discussion Forums. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, 6–12 January 2007; pp. 423–428.
38. Zhao, R.; Romero, O.J.; Rudnicky, A. SOGO: A Social Intelligent Negotiation Dialogue System. In Proceedings of the 18th International Conference on Intelligent Virtual Agents, Sydney, Australia, 5–8 November 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 239–246.
39. Segura, C.; Palau, À.; Luque, J.; Costa-Jussà, M.R.; Banchs, R.E. Chatbol, a Chatbot for the Spanish “La Liga”. In Proceedings of the 9th International Workshop on Spoken Dialogue System Technology, Singapore, 18–20 April 2018; Springer: Berlin/Heidelberg, Germany, 2019; pp. 319–330.
40. Ait-Mlouk, A.; Jiang, L. KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding Over Linked Data. *IEEE Access* **2020**, *8*, 149220–149230. [CrossRef]
41. De Bruyn, M.; Lotfi, E.; Buhmann, J.; Daelemans, W. BART for Knowledge Grounded Conversations. In Proceedings of the KDD Workshop on Conversational Systems Towards Mainstream Adoption (KDD Converse'20), San Diego, CA, USA, 24 August 2020; ACM: New York, NY, USA, 2020.
42. Zhao, X.; Wu, W.; Xu, C.; Tao, C.; Zhao, D.; Yan, R. Knowledge-Grounded Dialogue Generation with Pre-Trained Language Models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 3377–3390.
43. Kim, B.; Ahn, J.; Kim, G. Sequential Latent Knowledge Selection for Knowledge-Grounded Dialogue. *arXiv* **2020**, arXiv:2002.07510.
44. Gu, J.; Wang, Y.; Cho, K.; Li, V. Search Engine Guided Non-Parametric Neural Machine Translation. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 32. [CrossRef]
45. Dinan, E.; Roller, S.; Shuster, K.; Fan, A.; Auli, M.; Weston, J. Wizard of Wikipedia: Knowledge-Powered Conversational Agents. *arXiv* **2019**, arXiv:1811.01241.
46. Mzwri, K.; Turcsányi-Szabó, M. Chatbot Development Using APIs and Integration into the MOOC. *Cent.-Eur. J. New Technol. Res. Educ. Pract.* **2023**, *5*, 18–30. [CrossRef]
47. Kemp, S. The Latest Facebook Statistics: Everything You Need to Know. Available online: <https://datareportal.com/essential-facebook-stats> (accessed on 9 February 2023).
48. Kemp, S. The Latest Facebook Messenger Statistics: Everything You Need to Know—DataReportal—Global Digital Insights. Available online: <https://datareportal.com/essential-facebook-messenger-stats> (accessed on 23 March 2023).
49. Cho, J.; Garcia-Molina, H. Parallel Crawlers. In Proceedings of the 11th International Conference on World Wide Web, Honolulu, HI, USA, 7–11 May 2002; Association for Computing Machinery: New York, NY, USA, 2002; pp. 124–135.

50. Brin, S.; Page, L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Comput. Netw. ISDN Syst.* **1998**, *30*, 107–117. [CrossRef]
51. How Long Will It Take Google to Index My Site? Safari Digital. Available online: <https://www.safaridigital.com.au/blog/how-long-will-it-take-google-to-index-my-site/> (accessed on 17 November 2022).
52. Miklosik, A. Search Engine Marketing Strategies: Google Answer Box-Related Search Visibility Factors. In *Handbook of Research on Entrepreneurship and Marketing for Global Reach in the Digital Economy*; IGI Global: Hershey, PA, USA, 2018; pp. 463–485, ISBN 978-1-5225-6307-5.
53. Strzelecki, A. Website Removal from Search Engines Due to Copyright Violation. *Aslib Proc.* **2019**, *71*, 54–71. [CrossRef]
54. Miklosik, A.; Hlavatý, I.; Daňo, F.; Červenka, P. Google Answer Box Keyword-Related Analysis a Case Study. *Eur. J. Sci. Theol.* **2016**, *12*, 185–194.
55. Noy, N.; Gao, Y.; Jain, A.; Narayanan, A.; Patterson, A.; Taylor, J. Industry-Scale Knowledge Graphs: Lessons and Challenges: Five Diverse Technology Companies Show How It's Done. *Queue* **2019**, *17*, 48–75. [CrossRef]
56. Joshi, M.A.; Patel, P. Google Page Rank Algorithm and It's Updates. In Proceedings of the International Conference on Emerging Trends in Science, Engineering and Management, ICETSEM-2018, Karachi, Pakistan, 21–22 December 2018.
57. Baye, M.R.; De los Santos, B.; Wildenbeest, M.R. Search Engine Optimization: What Drives Organic Traffic to Retail Sites? *J. Econ. Manag. Strategy* **2016**, *25*, 6–31. [CrossRef]
58. Schütze, H.; Manning, C.D.; Raghavan, P. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008; Volume 39.
59. Salton, G.; Buckley, C. Term-Weighting Approaches in Automatic Text Retrieval. *Inf. Process. Manag.* **1988**, *24*, 513–523. [CrossRef]
60. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv* **2016**, arXiv:160304467.
61. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. Squad: 100,000+ Questions for Machine Comprehension of Text. *arXiv* **2016**, arXiv:160605250.
62. Cambria, E.; White, B. Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]. *IEEE Comput. Intell. Mag.* **2014**, *9*, 48–57. [CrossRef]
63. Biswas, M. Wit.Ai and Dialogflow. In *Beginning AI Bot Frameworks: Getting Started with Bot Development*; Apress: Berkeley, CA, USA, 2018; pp. 67–100. ISBN 978-1-4842-3754-0.
64. Kuo, Y.-F.; Wu, C.-M.; Deng, W.-J. The Relationships among Service Quality, Perceived Value, Customer Satisfaction, and Post-Purchase Intention in Mobile Value-Added Services. *Comput. Hum. Behav.* **2009**, *25*, 887–896. [CrossRef]
65. Maroengsit, W.; Piyakulpinyo, T.; Phonyiam, K.; Pongnumkul, S.; Chaovalit, P.; Theeramunkong, T. A Survey on Evaluation Methods for Chatbots. In Proceedings of the 2019 7th International Conference on Information and Education Technology, Aizu-Wakamatsu, Japan, 29–31 March 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 111–119.
66. Kazi, H.; Chowdhry, B.S.; Memon, Z. MedChatBot: An UMLS Based Chatbot for Medical Students. *Int. J. Comput. Appl.* **2012**, *55*, 1–5. [CrossRef]
67. Qiu, M.; Li, F.-L.; Wang, S.; Gao, X.; Chen, Y.; Zhao, W.; Chen, H.; Huang, J.; Chu, W. Alime Chat: A Sequence to Sequence and Rerank Based Chatbot Engine. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 498–503.
68. Bangor, A.; Kortum, P.; Miller, J. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *J. Usability Stud.* **2009**, *4*, 114–123.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.