Limiting online sexual Exploitation and Abuse Gender based on Underaged boys by Educating experts

D4.1 Chatbot

Funded by the European Union CERV programme under Grant Agreement No. 101049294. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Project

Acronym	**LEAGUE**
Title	Limiting online sexual Exploitation and Abuse Gender based on Underaged boys by Educating experts
Coordinator	Law and Internet Foundation
Reference	101049294
Type	CERV-PJG
Programme	CERV
Topic	CERV-2021-DAPHNE
Start	February 2022
Duration	24 months
Website	https://project-league.eu/
Consortium	Law and Internet Foundation (LIF), **Bulgaria** Centre for Research and Technology-Hellas (CERTH), **Greece** CESIE, **Italy** Institut Pravnih Znanosti, Raziskave in Razvoj na Podrocju Prava (OPHIZ), **Slovenia** SYNYO GmbH (SYNYO), **Austria**

Acknowledgement: Funded by the European Union CERV programme under Grant Agreement No. 101049294.	

Deliverable

Number	**D4.1**
Title	**Chatbot**
Lead beneficiary	CERTH
Work package	WP4
Dissemination level	Public (PU)
Nature	Report (RE) / Other (O)
Due date	31.01.2024
Submission date	03.04.2024
Authors	**Valadis Mastoras**, Msc and **Athanasios Mavropoulos**, Phd
Contributors	Charis Lagonidis
Reviewers	Denitsa Kozhuharova, LIF

Document history

Version	Date	Comments
0.1	1.12.2023	Valadis Mastoras, CERTH
0.2	10.12.2023	Charis Lagonidis, CERTH
0.3	20.12.2023	Valadis Mastoras, CERTH
0.4	25.1.2024	Athanasios Mavropoulos, CERTH
1.0	03.04.2024	Denitsa Kozhuharova, LIF

Executive Summary

This deliverable presents the development of the LEAGUE chatbot, a virtual bot application that was developed to support underaged boys on the topic of online sexual exploitation and abuse.

It is consisted of six main sections. These are, the *Introduction*, where a general overview of our work is demonstrated, the *Related work*, where our work is positioned in the recent literature, the *Chatbot implementation*, where the details of our implementation are presented, the *Evaluation*, where each one of the sub components of the LEAGUE chatbot are thoroughly evaluated and eventually, the *Stack of technologies* section, where all the leveraged technologies are presented.

In addition, a series of topics are highlighted in this deliverable. More precisely, a general overview of relevant chatbot technologies is presented, while also a thorough presentation of various such paradigms from the scientific literature. In addition, topics such as what data to use, what consists of a typical chatbot architecture, while also a series of sub components that are usually leveraged on such chatbot technologies, such as a Chitchat module and a Fallback Handler, are elaborated. Eventually, this deliverable also presents insights on ways to evaluate a chatbot, while also what technologies may be used for its development.

Contents

Table of Figures

Index of Tables

Acronyms & Abbreviations

Term	Description
BERT	Bidirectional Encoder Representations from Transformers
Bidirectional LSTM	biLSTM
DL	Deep Learning
FAQ	Frequently Asked Question
KB	Knowledge Base
LLM	Large Language Model
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
NLU	Natural Language Understanding
QnA	Question and Answer

1 Introduction

1.1 Chatbots in general

In contemporary times, there is a growing imperative to automate diverse facets of communication across various sectors, including Health Care, Marketing, Education, E-commerce, and more. This necessity arises primarily from the ever-expanding and accumulating volume of information that is characteristic of our era, which demands of efficient and automated methods to manage it effortlessly. Consequently, the literature consistently presents numerous examples of approaches aimed at automating these processes.

A notable form of such automation is the utilization of chatbots, serving as virtual assistants that facilitate human-computer interaction through Natural Language. These systems typically harness text or speech to address user queries and provide pertinent responses within a domain of interest. Thus, chatbots have greatly become a norm for various tasks over the past recent years, since with their aid, one can automate various time-consuming tasks (e.g. customer service, etc.), or handle more advanced communication topics (e.g. preliminary psychological support, etc.), all while reducing the required human effort and greatly eliminating possible human errors.

Chatbot systems manifest in various types, ranging from *rule-based* models, with predefined patterns and rules, to *retrieval-based* systems, which draw knowledge from existing sources to generate appropriate responses. In addition, *generative* chatbots have also been reported, capable of engaging in open-ended conversations through iterative learning and the creation of novel responses from scratch. At the same time, various types of supported knowledge have been documented, which serve as a basis for chatbot communication. This includes chatbots specializing in a *closed-domain* of knowledge, limited to aid with specific topics, as well as those operating within an *open-domain* of knowledge, enabling conversations on a wide range of generic topics.

These diverse chatbot types and knowledge domains can be implemented using different architectures, each emphasizing on specific aspects of communication. A typical chatbot architecture is typically comprised of three main modules: *Natural Language Understanding* (deciphering input queries and capturing user intent), *Dialogue Management* (managing conversation flow and state), and *Response Generation* (retrieving or generating relevant responses). However, even though the aforementioned modules of a chatbot architecture are quite frequently being leveraged, it is also quite often the case that only some of them might be leveraged, or they might be altered before being adopted, depending the necessities of a target task.

Therefore, the implementation of a chatbot is intricately tied to the task at hand and the available resources. Depending on the target task's goals, the choice of chatbot type, knowledge domain, and architecture may vary, highlighting in that way the adaptability and versatility of chatbot technology.

1.2 Our chatbot pipeline

In this work, we have coped with the task of creating a chatbot service that is capable of answering questions about various topics for a dedicated field. More specifically, our chatbot operates as a closed-domain system, dedicated to aid children that deal with issues related to the topic of online sexual exploitation and abuse. Essentially, the chatbot serves as an intermediary between children and a pre-existing Knowledge Base (KB), and it mostly functions as an automated Frequently Asked Questions (FAQ) responder.

In terms of its implementation, the chatbot (from now and on, the *League chatbot)* is consisted of a pre-stored KB and two integral processing modules. The pre-stored KB is a collection of pre-determined user questions, identified by human experts of our consortium as typical and frequently asked in such conversation settings. These questions were aligned to relevant answers, that were also manually constructed by the same experts. The first crucial module, the *NLU* module, interprets the meaning of input user queries. The second module, the *Response Management*, is responsible of controlling the conversation flow and deciding of which module or subcomponent to run. Eventually, the *Answer Retrieval* module (vastly known as Language Generation), is responsible for retrieving and presenting an appropriate response, based on the input query.

To execute this process, we integrated a *semantic similarity search* component, typically applied on settings where texts are being compared in terms of their semantics. More precisely, this component automatically identifies the degree of semantic relatedness between an input user query and all the pre-determined questions in our KB. Having identified the most semantically related question in the KB, the chatbot retrieves and presents the answer that is aligned to that identified question, as a response to the user through the Answer Retrieval module. An illustration of that procedure can be seen on Figure 1.

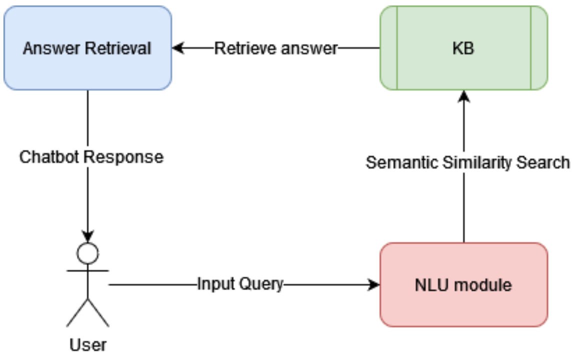


Figure 1. The overall pipeline of the chatbot conversation procedure.

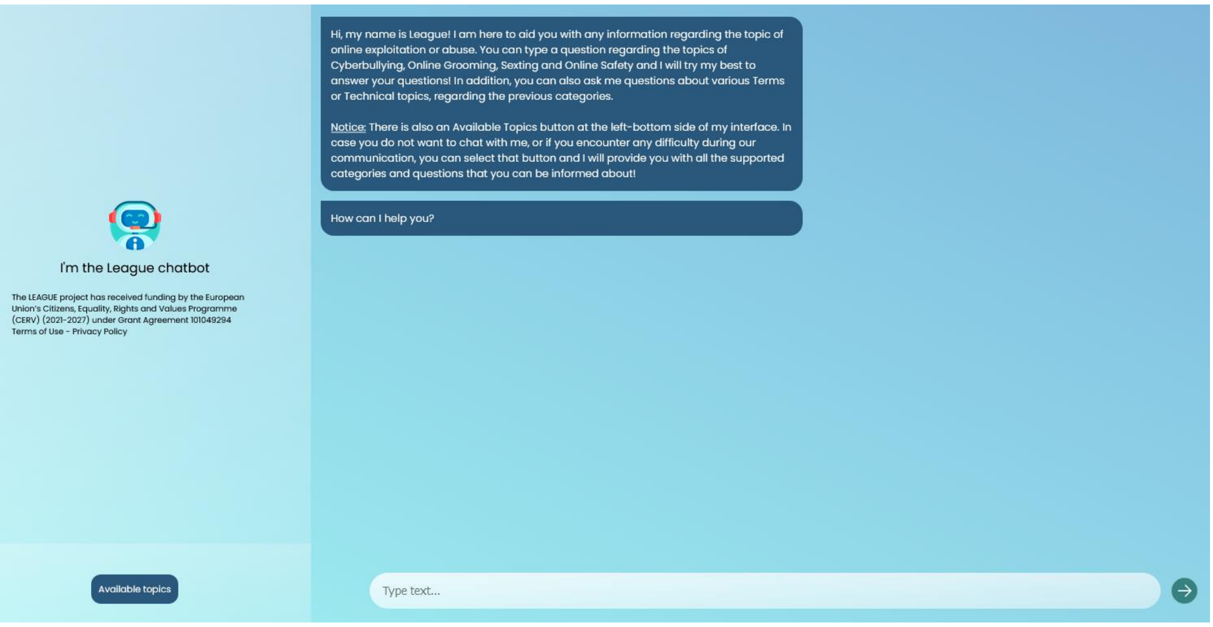


Figure 2: The chatbot interface.

These functionalities have been implemented within a fully functional web application, complemented by additional components aimed at enhancing the chatbot's fluency and overall performance. The chatbot interface can be seen on Figure 2. The interface, along with all the associated components, will be subsequently elaborated extensively. The final version of the LEAGUE chatbot can be accessed on: https://project-league.eu/.

Following in this report, on Section 2, we will initially present a brief enumeration of some of the most frequently adopted approaches for the task in hand, so that the reader might be aware of the basic implementation directions that exist on this field. Subsequently, on Section 3, we will thoroughly present our chatbot implementation, while also on Section 4, we will dedicate a chapter for the evaluation of our work. On Section 5, we will mention all the technologies that were leveraged to create our application, before concluding our work on Section 6.

2 Related work

In recent years, there has been a notable proliferation of chatbots across various topics, which have effectively addressed with different sorts of information. These chatbots have found application in diverse fields, including among others, education (e.g., [1] and [2]), customer assistance [3], or healthcare [4]. Moreover, chatbots have also been introduced on more sensitive topics, such as online child abuse. A noteworthy instance in this context may be the work of [5], who introduces a specialized chatbot framework to handle child abuse-related inquiries within the realm of online child protection. Following the latter topic, our work is also targeted on the subject of online child abuse.

Besides their application across diverse topics, chatbot technologies may be broadly classified into two main categories, those of *open-domain* and *closed-domain* ones. Starting with the open-domain ones, these chatbots have been mostly engineered to participate in conversations spanning a wide array of topics, including subjects such as generic-purpose daily life conversation [6] or emotional support [7], etc. Such chatbots are extensively explored in the literature, with a prevailing recent trend involving the widespread adoption of Large Language Models (LLMs) [8] to generate contextually relevant responses. However, it is important to note that such chatbot types may be susceptible to the so-called *hallucinations*, a phenomenon that is tightly depended on such generative approaches, which raises concerns wherein the chatbot generates seemingly realistic but inaccurate information [9]. Therefore, a key challenge associated with the open-domain chatbots is their lack of control on the final output, potentially compromising accuracy, especially on domain-specific topics of a more sensitive nature, such as online child abuse, which may demand a higher level of output control.

On the other hand, closed-domain chatbots are designed to be specialized on more specific topics or industries, offering precise and targeted responses within predefined domains. Some indicative examples of closed-domain chatbots may be the one of [10], where a pre-trained chatbot model was proposed, which was fine-tuned on six distinct domains, ranging from e-commerce to banking, while also a controllable response selection approach was leveraged. Furthermore, the proposal of [11] may also be an indicative example of a closed-domain chatbot for the Chinese movie domain. Therefore, in contrast to their open-domain counterparts, closed-domain chatbots mostly excel in domain-specific knowledge. Consequently, the choice between open-domain and closed-domain chatbots becomes a matter of necessity rather than complexity, as closed-domain chatbots prove to be more suitable for applications demanding precision and control within specific domains.

In our case, we will follow the paradigm of a closed-domain chatbot, which leverages the pre-defined knowledge of pre-trained embeddings to answer in an appropriate manner. In more details, we find two literature examples particularly pertinent to our study. These are the closed-domain chatbot subcategories that concern *FAQ type chatbots* and mostly those that employ *Text Semantic Similarity* calculations. The FAQ type chatbots hold relevance for our work due to the structure of our data, which consists of pairs of potential user questions and their corresponding answers (we will delve into the details of this on a subsequent chapter). The Text Semantic Similarity approach is significant because it serves as the primary implementation module that facilitates the communication with our chatbot. Therefore, before delving into the intricacies of our implementation, we may also introduce some notable examples in the literature.

Starting with the FAQ chatbots, two notable examples may be those of [12] and [13]. In the former approach, a domain-specific FAQ chatbot for Thai language was developed, which classified any input text on a pre-determined group of questions, utilizing a Long Short-Term Memory (LSTM) model, before retrieving a relevant response for that question group. In the latter approach, a university chatbot that could deal with FAQs across various domains based on a Deep Learning (DL) model was designed, which was trained on hundreds of different user intents and their possible responses. However, even though DL models may appropriately serve such tasks, those typically require a vast amount of training data to function properly. Therefore, another noteworthy instance of FAQ implementations may be the one exemplified in [14], which did not leverage a DL approach. That proposal is also the most closely related to our work. More precisely, in [14], a chatbot was developed that employed a knowledge database, structured as two-dimensional string arrays, enabling in that way for interpretation and response of frequently asked user questions effectively. More precisely, given a user input, that chatbot would analyse it through a pattern-matching approach and based on that, the FAQ database would be accessed to retrieve predefined patterns that would serve as chatbot responses. This last approach holds great relevance to ours, even though the pattern-matching approach is rather superficial compared to the approach we leverage.

Eventually, in the realm of literature employing Text Semantic Similarity calculations, numerous chatbot approaches have been proposed, with [15], [16] and [17] standing out as some of the most indicative examples. Specifically, [15] implemented a handcrafted measure for Text Semantic Similarity tailored for Chinese, incorporating a typical pattern-matching technique to align input sentences to stored sentences which were further associated with chatbot responses, while also [16] applied a similar approach, though for Indonesian. On the other hand [17] proposed a chatbot for college information acquisition, which was created following a process that involved the extraction of keywords from input questions, while also the employment of an algorithm to assess sentence similarity between the input question and a predefined question base, aligning it with pertinent answers based on a target similarity confidence threshold. Inspired from the aforementioned, we have also followed a similar Text Semantic Similarity methodology, leveraging though more advanced pre-trained embeddings, instead of the aforementioned superficial approaches.

3 Chatbot implementation

Having introduced the overall pipeline of the chatbot, as well as positioned it among the recent literature, we will now elaborate the specifics of our work in a more thorough manner.

We will start by presenting the core of our work, which is the data that consists the KB of our chatbot and the way those were processed to form a useful resource. Subsequently, we will continue by presenting each one of the components of our chatbot architecture in more details.

3.1 Data

As it was already mentioned, this chatbot belongs to the vast category of closed-domain, informative chatbots. Therefore, the data that are being leveraged are essential, since those consist the knowledge pool that the chatbot leverage to perform its core actions.

In our work, the data-in-use were collected by human experts of our consortium based on an analysis of the most informative and most frequent user inquiries, concerning the topic of online sexual exploitation and abuse, which were retrieved from T3.1 and T4.1 of this project. An initial set of such questions were collected, organized on categories and aligned with a manually constructed answer by the same human experts. The collected categories were those of *Cyberbullying*, *Online Grooming*, *Sexting*, *Technical-Cyber*, *Online Safety*, as well as a category dedicated to *Terminology*. An example of the initial Question and Answer (QnA) data, with instance questions pooled from the Cyberbullying category, can be seen on Figure 3.

Questions	Answers
Could you please explain to me what cyberbullying means?	Cyberbullying is when someone bullies or harasses others on the internet and other digital spaces. Cyberbullying includes sending, posting, or sharing negative, harmful, false, or mean content about someone else. It can include sharing personal or private information about someone else causing embarrassment or humiliation.
How do I know if I'm being cyberbullied?	Good question. Nowadays it is hard to see the difference between someone joking ("messing") with you or bullying you. The key to determine if you are subject to bullying is to answer the following questions – first how the situation made you feel (e.g., did it make you feel uncomfortable, down, unhappy or has offended you), and whether the other person's behavior continued even after you explained how you felt and asked the person to stop. If the answer to the above questions is "yes", then the situation is most likely related to bullying and moreover to cyberbullying if it has happened in an online environment.

Figure 3. Two example QnA pairs from the initial data of the KB, for the cyberbullying category.

Having such data from our partners, we have performed a series of pre-processing steps to render them valuable for our purpose. More precisely, we have manually paraphrased in an exhaustive manner, each collected question of the base data. The paraphrasing was carefully executed to preserve the core meaning of each question, while enhancing both the sentence structure (with a variety of syntactical forms) and the richness of the vocabulary. Each one of the paraphrased questions was aligned with a corresponding category and a relevant answer, based on the initial, seed questions that were used as a basis, constructing in that way an extensive set of paraphrased Question and Answer (QnA) pairs. An example of the final data form, where a subset of such paraphrased questions is demonstrated, can be seen on Figure 4.

Eventually, it should be mentioned that, strong considerations where held concerning whether the chatbot should also be implemented on other languages than English, and therefore acquire relevant QnA pairs for other such languages. However, it was decided not to follow that path, because the core functionality of the chatbot heavily depends on various pre-trained NLP models (as it will more thoroughly elaborated subsequently). Those models are usually pre-trained on a huge series of training data and we use the knowledge of that previous training, for our task. However, even though such models may exist in an abundance for English, that is not the case for other languages. In that sense, given the current development approach, low-resource languages like Greek or Bulgarian proved to be quite poor both in terms of model availability and model effectiveness.

Paraphrased Question	Answer
I have started receiving threatening messages and I feel unsafe. What should I do?	No matter what the situation that provokes these actions is, such behavior is inappropriate and wrong, and is classified as cyberbullying. Preferably, you should block the individual/person/user/player who is texting you and report it to the group platform/ game owners. If the situation has escalated further and you feel overwhelmed by the attitude of the individual/person/user/player, you should share with your parents and discuss your feelings further with a psychologist if necessary. For the future, it is also important to set appropriate settings in the (game) platform in relation to who can contact you and limit the available options – you can ask your parents if you are unsure how to set these up or look for video tutorials on the internet.
I have received abusive messages, what should I do?	
Someone has sent me some very mean and threatening messages and I am wondering what I can do	
I was online and I started getting threatening messages from people I do not know. What to do omg?	

Figure 4. An example of QnA pairs, paraphrased, from the initial data of the KB, for the cyberbullying category.

3.2 Architecture

Having presented the core KB of the chatbot, the next step of our report will concern the architecture and the implementation details of it. In more details, the LEAGUE chatbot is consisted of 3 main modules and an additional, experimental one. Those modules are, correspondingly, the *Core module*, the *Chitchat module*, the *Fallback handler* and the *Multi-Turn module*.

Starting with the Core module, that is the one that handles all the user questions that are relevant to the knowledge supported in the KB. Subsequently, the Chitchat module, is an additional module that handles a variety of daily life topics, aiming to enhance user communication. Moreover, the Fallback Handler is a component that is responsible of handling any user question that is either out-of-scope in terms of the supported knowledge, or incomprehensible by any means. Eventually, the Multi-Turn module is an experimental component that was developed to render the chatbot more context-aware by maintaining a recent conversation history. All these modules have been developed as separate components that function in an interconnected manner.

In more details, given an input question from the user, the Core module is responsible of identifying whether that question can be handled with the existing KB. If that is not the case, then the Chitchat module takes over and examines whether the input question refers to a daily-life topic. Subsequently, if none of the aforementioned components is able of coping with the input question, then the Fallback handler takes over, to handle accordingly this case. Eventually, along with the Fallback handler, the Multi-Turn module is also leveraged, which examines whether the user might request information that already exists on the previous conversation history. All the aforementioned are also illustrated on Figure 5. A more detailed overview of each specific component, will be presented subsequently.

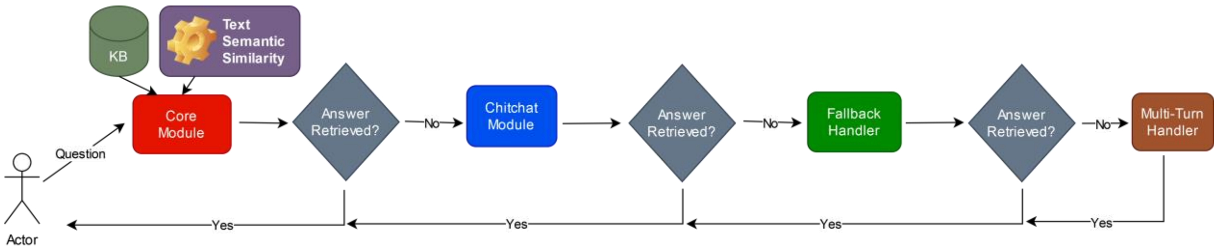


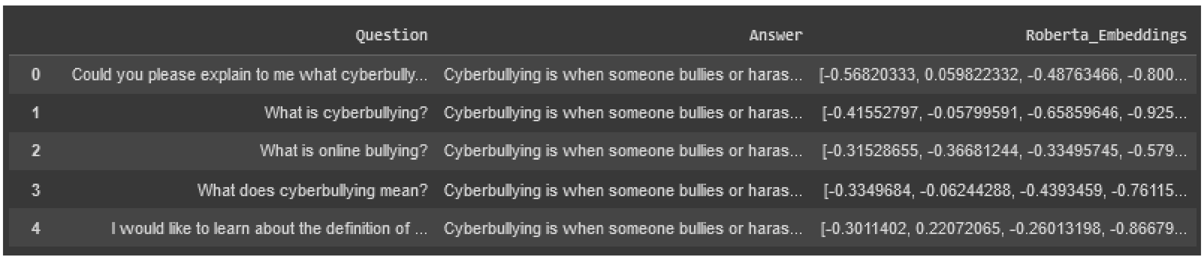
Figure 5. The chatbot pipeline.

3.2.1 Core Module

As already mentioned, the first component of the LEAGUE chatbot that processes the input question, is the Core module. That module, forms the NLU part of the chatbot and it was developed using a *Text Semantic Similarity* approach, based on which, each input question is examined in terms of semantic relatedness against every question stored on the KB, to identify the stored question that is the most similar to the input one.

More precisely, to develop this module we have explored the usefulness of pre-trained embeddings, which are quite frequently leveraged to extract the semantics of a given text and store them in a vectorial representation, so that they may be reused for a future goal. Usually, such embeddings are learned for a certain task and they are subsequently re-used to solve another. We have leveraged a well-known transformers-based model that has been a go-to model for a variety of Natural Language Processing (NLP) tasks, while also when utilizing pre-trained embeddings, the Bidirectional Encoder Representations from Transformers (BERT) model [18] and more precisely, an optimized sentence-level version of it, namely *Roberta* [19]. Therefore, with the aid of the Roberta model, we retrieved the vectorial representations of each one of our stored questions. can be seen on Figure 6.

Having those vector representations, the LEAGUE chatbot converts any input user question on its vectorial representation, utilizing the same pre-trained embeddings, and compares it against the vectorial representations of all the stored questions in the KB. A measure called cosine similarity is leveraged, to calculate the angle between the input question vector and each one of the pre-stored question vectors. In that way, the semantic similarity of those is calculated based on a standard similarity threshold, above which, it is assumed that an adequate semantic similarity exists. Eventually, the stored question that has the highest cosine similarity score against the input user question, is retrieved from the KB and the answer that is aligned to that stored question, is fetched and demonstrated as a response to the user. That process forms the response management part.



	Question	Answer	Roberta_Embeddings
0	Could you please explain to me what cyberbully...	Cyberbullying is when someone bullies or haras...	[-0.56820333, 0.059822332, -0.48763466, -0.800...
1	What is cyberbullying?	Cyberbullying is when someone bullies or haras...	[-0.41552797, -0.05799591, -0.65859646, -0.925...
2	What is online bullying?	Cyberbullying is when someone bullies or haras...	[-0.31528655, -0.36681244, -0.33495745, -0.579...
3	What does cyberbullying mean?	Cyberbullying is when someone bullies or haras...	[-0.3349684, -0.06244288, -0.4393459, -0.76115...
4	I would like to learn about the definition of ...	Cyberbullying is when someone bullies or haras...	[-0.3011402, 0.22072065, -0.26013198, -0.86679...

Figure 6. A snapshot of the final form of our KB

3.2.2 Chitchat module

Besides the core communication procedure, an additional module, namely the Chitchat module, was developed to render the conversation more pleasant and fluent for the user. This module is responsible of handling a variety of daily-life and generic-purpose topics, that may be usual on any conversation setting.

To develop it, a series of models were trained on a custom dataset and they were evaluated on how well they were able to identify a target chitchat intent, given a relevant input question. To train our models, the data provided by Microsoft[1] were leveraged, due to their open nature (MIT license) and their relativity to our task. Those data were then pre-processed and only a series of useful data categories were leveraged, while also, the instances in those categories were further tailored to keep only those that were suitable for our task. Furthermore, the training instances were manually enriched with examples that were manually created or retrieved from online resources. The final training data were in the form of input questions, labeled with a relevant chitchat intent. A final number of 27 different intents and a sum of around 3000 total of instances existed on the dataset. The intents, along with more details, can be more thoroughly seen on Appendix I.

Having our dataset prepared, we trained a series of different models to evaluate which one was the most relevant for our task. Among all the relevant candidates, the BERT model demonstrated the greatest performance, and was therefore adopted. A detailed overview of the models and their training results will be presented on Section 4. In the end, the trained model was able of accepting an input question and classifying it to one of the supported chitchat intents. Moreover, a very high confidence threshold was set, so that the model would be forced to be as certain as possible when making predictions, since the Chitchat module should only take over from the Core module on cases of absolute certainty. Having identified the target intent, this module accesses a dedicated KB that was manually constructed for it, in which each intent is aligned with a series of possible chatbot responses, all manually handcrafted. A response is pooled from these and is presented as a response to the user.

3.2.3 Fallback handler

Continuing with the presentation of the main modules, the third module of our implementation is the Fallback Handler, which is responsible for cases where the chatbot is not able of understanding an input user question. More precisely, the first time that the LEAGUE chatbot will not understand an input question, either because it is not supported in the KB or because the input question is incomprehensible in any other way, the chatbot will return a message, asking the user to rephrase. Then, in case of an incomprehensible input question for two times in a row, the Fallback handler will pause the conversation thread and the user will be forced to manually select among all the supported topics, which will be displayed as a grid of buttons. In that way, whenever the chatbot is not able of handling the communication process, the Fallback handler may aid for a seamless functioning of the service. An illustration of the Fallback handler can be seen on Figures 7 and 8.

In a similar sense, a button exists at the left bottom corner of the chatbot interface (namely, *Available Topics*), which can be utilized at any time of the conversation, to acquire immediate access on all the supported topics.

[1] https://github.com/microsoft/botframework-cli/blob/main/packages/qnamaker/docs/chit-chat-dataset.md

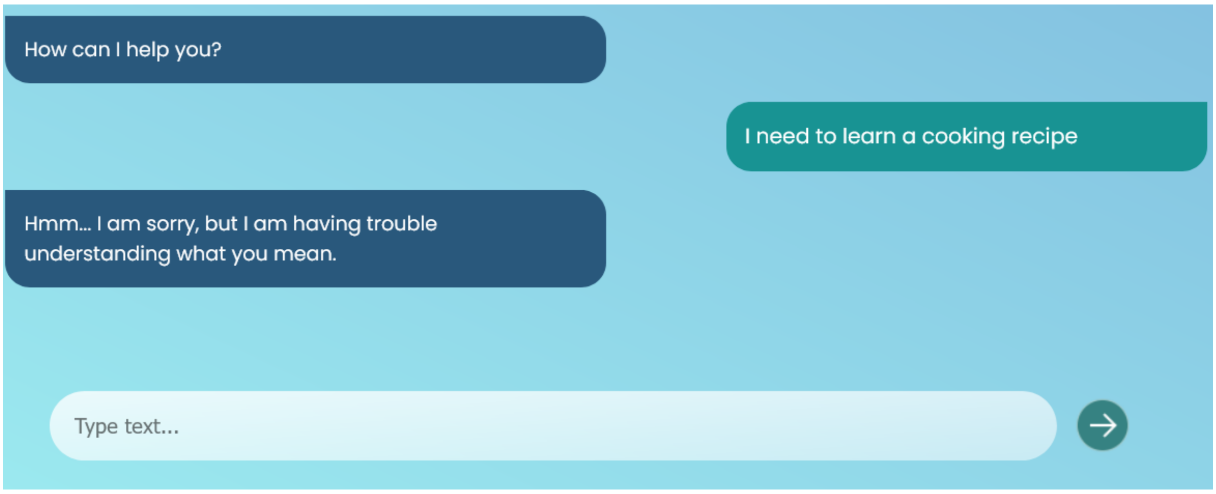


Figure 7. A snapshot of the Fallback Handler for the first time of incomprehensible question.

Figure 8. A snapshot of the Fallback Handler for two consecutive incomprehensible questions.

3.2.4 Multi-turn module

Conclusively, as it was already mentioned, besides the Fallback handler, another component, namely the Multi-Turn module, was also developed to render the chatbot more context-aware. This module is a rather experimental one, which was mainly developed to enhance the chatbot communication ability to any extent it would be possible.

More precisely, this component is responsible of maintaining a history of the past conversation with the user. Then, whenever a user types an incomprehensible input question for two times in a row, before entering the previously mentioned Fallback handler, this module takes over and proposes a possible topic of interest to the user, based on the previous conversation history. Similarly to the Core module, this module also leverages a Text Semantic Similarity approach to perform its actions. In fact, while the aforementioned, default communication process is being executed, this module keeps a history of the past ten questions that were asked by the user. That number has been empirically selected and verified, based on the hypothesis that after a certain number of dialogue turns, it may be safely assumed that the user will shift the dialogue topic and thus it would be irrelevant to search for user questions that were asked before it.

In more details, whenever an incomprehensible user question is given to the chatbot for two times in a row, before entering the Fallback handler, the Multi-turn module examines for the semantic relatedness of that input question against the previously asked questions, stored on the conversation history. The semantic similarity of this module is performed in a looser manner than the one of the Core module, utilizing a much smaller similarity threshold, to extend the possibility to capture even a vague semantic relatedness. Then, in case a semantic similarity is observed against any of the past user questions in the history, a message will appear to the user asking whether the identified previously asked question belongs to the target subject of interest. In case of a negative response from the user, the Fallback handler will be performed as already demonstrated. Otherwise, the previously asked question that is identified, will be leveraged to isolate its most similar question from the KB and return the aligned response as an answer to the user, while also resetting the conversation thread back to its default state. An illustration of the Multi-turn module may be seen on Figure 9.

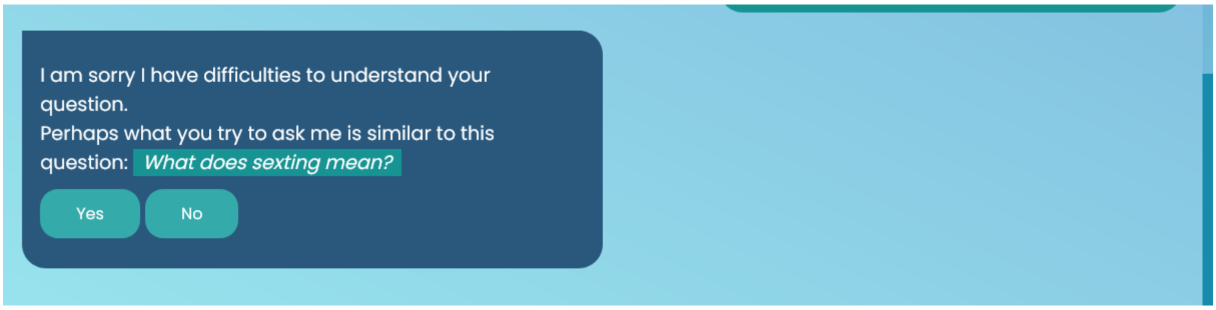


Figure 9. A snapshot of the Multi-Turn module.

4 Evaluation

Concerning the evaluation part, we were limited on only partially examine the subcomponents of our work, since the results of a chatbot conversation are rather difficult to be measured. Thus, we have considered that the chatbot should mostly be evaluated by the end users, through their actual conversation experience. Until that point, we may mostly present some initial insights on specific parts of the League chatbot, which may be indicative of its total behaviour.

4.1 Core module

Starting with the Core module, we have tested both the Glove[2] pre-trained embeddings for various dimensions (e.g., 50, 100 and 300), while also the Roberta pre-trained embeddings, to decide which embeddings might be leveraged for our Text Semantic Similarity procedure. The reason for that, was to evaluate both typical word-based embeddings while also the more recent, sentence-based ones, which can further capture sequential and contextual information.

More precisely, we have manually constructed an initial evaluation file, which contained base questions retrieved from the QnAs of our KB, each manually paraphrased on a superficial manner. For each base question, in each topic category, a total of five paraphrased questions were crafted. The final evaluation file contained 118 instances in total. Then, we evaluated all the handcrafted questions against their base questions for a range of cosine similarity thresholds (e.g., 0.5, 0.6, 0.7, 0.8 and 0.9). Our evaluation concerned which of the pre-trained embeddings and which of the thresholds would manage to acquire the best results, meaning to semantically match in the most appropriate manner, each of the paraphrased questions to their corresponding base question. Detailed statistics can be seen on Tables 1, 2, 3 and 4.

Among the evaluated embeddings, we can see on Tables 1, 2 and 3 that the Glove embeddings have mostly demonstrated a stable behaviour, with excellent Precision but rather low Recall for all the examined dimensions. On the other hand, on Table 4, we can see that the Roberta sentence embeddings have also demonstrated an excellent Precision, while also a remarkable Recall, at least until the threshold of *0.7*. Considering the nature of this task, we may claim that Recall is a more valid measure for our goal, since it can better measure how well the model can capture the positive instances, even with a compromise of making some extra False Positive predictions. Since this part of our work, as already stated, is quite vague to be evaluated, and since this initial evaluation part served only as a preliminary baseline to retrieve insights on the implementation path to choose rather than a concrete evaluation approach, we have decided to mostly rely on the Recall results. Therefore, we have adopted the Roberta sentence embeddings with a threshold of 0.6, which demonstrated the best compromise between Precision, Recall and Confidence.

2 https://nlp.stanford.edu/projects/glove/

Table 1.

Threshold	Precision	Recall	F-score
0.5	1.0	23.12	37.56
0.6	1.0	23.12	37.56
0.7	1.0	23.12	37.56
0.8	1.0	23.12	37.56
0.9	1.0	22.9	37.35

Table 2.

Threshold	Precision	Recall	F-score
0.5	1.0	28.01	43.7
0.6	1.0	28.01	43.7
0.7	1.0	28.01	43.7
0.8	1.0	28.01	43.7
0.9	1.0	27.93	43.6

Table 3.

Threshold	Precision	Recall	F-score
0.5	1.0	33.79	50.51
0.6	1.0	33.79	50.51
0.7	1.0	33.79	50.04
0.8	1.0	33.30	49.96
0.9	1.0	28.33	44.16

Table 4.

Threshold	Precision	Recall	F-score
0.5	1.0	81.10	89.56
0.6	1.0	75.89	86.29
0.7	1.0	56.43	72.14
0.8	1.0	18.07	30.62
0.9	1.0	0.2	3.9

4.2 Chitchat module

For the Chitchat module, we have trained and evaluated 4 different models to examine which one performs better for our needs. More precisely, we have firstly trained a rather simplistic linear model with Bag-Of-Words embeddings. This type of embeddings represent any text by counting the frequency of words in a document, treating in that way, each word independently in a superficial manner without taking word sequential information under consideration. Moreover, similarly to the Core module, we have also experimented with a Bidirectional LSTM (biLSTM) model, which is capable of capturing bidirectional information in sequential data. Pre-trained Glove embeddings of 300 dimensions have been also leveraged. Eventually, we additionally trained a BERT model due to its ability to capture contextual information, in addition to all the aforementioned aspects.

This task was treated as a multi-class classification problem. Our data were consisted of 27 classes (see also Appendix I) with a total of around 3000 instances and to evaluate our results, typical classification metrics were leveraged, such as *Accuracy*, *Recall*, *Precision* and *F1-score*. Detailed statistics are demonstrated on Tables 5, 6 and 7. As it can be seen, the BERT model has demonstrated the best behavior with an F-score of 93%.

In addition, we have manually constructed an evaluation dataset, similarly to what has already demonstrated on Section 4.1. More specifically, for each one of the 27 classes, we have randomly pooled 5 examples from the data that the models were already trained on, and we have also manually crafted another 5 paraphrases of them, which were unknown to the models during training. Then, each one of the aforementioned models was evaluated on those evaluation examples. Among all models, BERT again demonstrated the greatest performance with 84% F-score.

Therefore, the BERT model was selected for this part of our task. Furthermore, we have also tested different architectures and hyperparameters to evaluate whether we may get any better performance with any other of them. The best results were given with the *cased model* and a hidden size of 768. The model was trained for 20 epochs, using the typical hyperparameters that are leveraged on such settings.

Table 5. BOW chitchat results

Evaluation Setting	Accuracy	Precision	Recall	F-score
Model Training	0.85	0.83	0.82	0.82
Evaluation Dataset	0.65	1	0.65	0.75

Table 6. biLSTM chitchat results

Evaluation Setting	Accuracy	Precision	Recall	F-score
Model Training	0.8	0.82	0.73	0.75
Evaluation Dataset	0.44	1	0.44	0.6

Table 7. BERT chitchat results

Evaluation Setting	Accuracy	Precision	Recall	F-score
Model Training	0.96	0.94	0.93	0.93
Evaluation Dataset	0.72	1	0.72	0.84

4.3 Fallback handler and Multi-turn module

Concerning the Fallback handler, a two-fold evaluation was conducted, concerning both the button-based and the text-based handlers, which were previously presented. More precisely, we have firstly evaluated the button that is linked to this handler (i.e. *Available Topics* – see Section 3) in an exhaustive manner, to validate that every component of its interface would function properly and with the correct content. Moreover, the same evaluation approach was applied for the text-based version of this handler, where all the components of the target interface were thoroughly evaluated both in terms of correct response and response time. The response time of the Fallback Handler did not exceed of *5 seconds* for any incomprehensible question.

Eventually, as long as it concerns the multi-turn module, that was only partially evaluated with superficial conversation scenarios to evaluate that it functions correctly. There was no extensive evaluation of this part, since it is only considered as an experimental enrichment of the current work.

4.4 Chatbot application

Conclusively, the LEAGUE chatbot was also evaluated as a fully functional application. More precisely, an exhaustive application testing to evaluate its response time was conducted, which remained stable in a mean time of *2 seconds*. Subsequently, each module was tested both concerning its functionality as a separate module, while also in interaction with the rest of them, as a complete application.

Besides the aforementioned, we have also retrieved great feedback from users during our international and national workshops, which lead to significant changes both on the content of the chatbot and on various aspects of its behaviour, helping us in that way to improve its current state. Similarly, the feedback that was obtained for the chatbot response time and for the overall interface of it, was positive, which may also be a validating point at some degree.

However, even though all the aforementioned might be indicative of the chatbot's current condition, those are unfortunately insufficient to properly evaluate it in depth, since the real performance of it, is strongly related to the overall communication procedure. Thus, at its most, the chatbot will be properly and more thoroughly evaluated when disseminated to the end users, who will have the chance to exhaustively communicate with it and aid us with possibly relevant feedback.

5 Stack of technologies

The primary programming language utilized in the development of the LEAGUE chatbot was Python. Specifically, we made use of prominent NLP libraries, including Pandas[3] and NLTK[4]. Additionally, we leveraged the PyTorch[5] and Tensorflow[6] frameworks to either develop or employ our DL models.

For the construction of the chatbot's web application, we opted for the Flask[7] framework. The back-end of the application was implemented using Python, while the front-end was crafted with HTML5, CSS3, and JQuery. To enhance the design and functionality, we also incorporated the Bootstrap[8] framework.

Given the limited extent of our base data, there was no imperative need to build a dedicated database for our application. Consequently, our data were primarily stored in files of the *.csv* and *.json* formats, and we managed them using relevant Python libraries. This approach allowed us to efficiently handle and manipulate the data within the context of our chatbot application.

3 https://pandas.pydata.org/
4 https://www.nltk.org/
5 https://pytorch.org/
6 https://www.tensorflow.org/
7 https://flask.palletsprojects.com/en/3.0.x/
8 https://getbootstrap.com/docs/5.3/getting-started/introduction/

6 Conclusion

In this WP, we have introduced and developed a novel chatbot application, namely the LEAGUE chatbot, which aims on assisting child users, potentially harassed or bullied, on topics related to children online sexual exploitation and abuse.

More specifically, given a set of topic-related data, in the form of possible user questions aligned with an answer, we have constructed a relevant KB that can support all the chatbot procedures. Based on that source, the chatbot can identify and retrieve any necessary information and communicate in a proper manner. Our work is mostly focused on the NLU part of a typical chatbot architecture, meaning that we have aimed on developing a system that will be able of properly understanding the meaning of any input query, while also constraining the chatbot responses to a set of pre-defined answers, due to the sensitivity of the topic in hand and the fragility that a child in such situations may carry. This chatbot acts mostly as an informative agent, where given an input question by a user it is able of deciphering its meaning, before aligning it to a relevant question in our KB and retrieve an appropriate answer to demonstrate it as a response. Moreover, a set of additional modules have been implemented to enhance the chatbot’ s behaviour, either by rendering the communication more fluent and context-aware, or by handling any incomprehensible or unsupported question. Eventually, a fully functional application interface was implemented for users to access and communicate with the chatbot easily.

Conclusively, each of the chatbot modules have been thoroughly tested, while also the application as a whole was evaluated. Those preliminary tests have already indicated that our chatbot system might be in a rather sufficient state. However, the main evaluation part of this work will be conducted in the future, mainly through the user experience when it will be publicly released.

7 Appendix I

Intent	Example Input question	Example Chatbot response
feel_lonely	I wish I had a friend	Well, I am here if you need me. If I could provide you with any help regarding the topic of "Online sexual Exploitation and Abuse", I would be glad to.
you_dumb	Everything you say is so dumb	I am sorry about that. Unfortunately, my capabilities are rather limited for the time being. You can always select the "Available Topics" button to learn about the topics I support.
leaving	I have to log out	Bye!
are_you_fake	Are you a robot or a person?	I am a chatbot who was created by researchers. My name is League and my aim is to help you with information regarding the topic of "Online sexual Exploitation and Abuse"
can_you_talk	Can you speak to me?	Sure! I am always happy to chat and guide you for any problem that is relevant to the topic of "Online sexual Exploitation and Abuse"
can_you_dance	Can you cook dinner for me	Unfortunately, that is not one of my talents. I can mostly answer your questions about the topic of "Online sexual Exploitation and Abuse".
pissed_off	I'm so pissed off right now	I am so sorry to hear that. Do not hesitate to ask me something regarding that if you want to, and I will try my best to answer it.
repeating_yourself	You keep saying the same stuff	I am terribly sorry if I am repetitive. I only carry some pre-defined answers, regarding the topic of sexual Exploitation and Abuse"
what_you_think_of_me	What kind of person do you think I am?	You seem quite nice! However, my goal is mostly to help you with the topic of "Online sexual Exploitation and Abuse". Let me know if you need anything regarding that!
hey	Hey there	Hello there
stop_talking	I don't want to hear anything from you	Sure, I am sorry. If you need anything else, I will be around.
who_made_you	Did someone create you?	Researchers have created me to help you with the topic of "Online sexual Exploitation and Abuse". If you want to learn, more you can check the information under "Consortium" on the site navigation bar. You do not have to worry for anything concerning that though. Our conversation is totally private and the only aim is to provide you with information regarding the topic of "Online sexual Exploitation and Abuse"

that_is_correct	that is true	I am glad!
how_are_you	How are you doing?	I am great, thanks
boring	Why are you so boring?	I am sorry, but my capabilities are limited. I can only provide you with information regarding the topic of "Online sexual Exploitation and Abuse".
confused	What's that mean?	If something that I said is not clear to you, you could perhaps try to ask me again in a different and more specific manner. Otherwise, you can always choose the “Available Topics” button to select the topic of your interest. I am sorry in any case!
how_you_assist	What is your purpose?	I am here to provide you with information regarding the topic of "Online sexual Exploitation and Abuse".
who_are_you	So what is your name?	I am the LEAGUE chatbot. Nice to meet you!
thank_you	Wonderful, thanks!	You're welcome!
hang_on	Be right back	Sure, I'll be here
nice_meet_you	It’s nice to meet you	It's nice to meet you as well
I_like_you	I like you	That's really nice of you. Thank you!
hug	Can I get a little hug?	I am really sorry but that's not something I can do.
returned	Hey I'm back	Hello again
be_my_friend	You are my best friend	That is so nice of you, thanks! Unfortunately, my capabilities are rather limited and I cannot support a friendship. I can only answer questions regarding the topic of "Online sexual Exploitation and Abuse". Sorry!
you_hate_me	I think you hate me	That is not true and I am sorry if I gave you a wrong impression! My only aim is to answer your questions regarding the topic of "Online sexual Exploitation and Abuse"
do_you_like_me	Do u like me?	You seem nice! However, I am just a bot and I do not have personal feelings.

8 References

[1] F. Colace, M. D. Santo, Marco Lombardi, F. Pascale, A. Pietrosanto, Saverio Lemma. (2018). Chatbot for e-learning: A case of study. *International Journal of Mechanical Engineering and Robotics Research, 7*(5), 528–533.

[2] Bhavika R. Ranoliya, Nidhi Raghuwanshi, Sanjay Singh. (2017). Chatbot for university related FAQs. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, (pp. 1525–1530).

[3] Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou. (2017). SuperAgent: A Customer Service Chatbot for E-commerce Websites. *Proceedings of ACL 2017, System Demonstrations* (pp. 97–102). Vancouver, Canada: Association for Computational Linguistics.

[4] S. Divya, V. Indumathi, S. Ishwarya, M. Priyasankari, and S. K. Devi. (2018). A self-diagnosis medical chatbot using artificial intelligence. *Journal of Web Development and Web Designing, 3*(1), 1-7.

[5] Pei Wang, Zhen Guo, Lifu Huang. (2021). SERI: Generative Chatbot Framework for Cybergrooming Prevention. *The First Workshop on Evaluations and Assessments of Neural Conversation Systems (EANCS).*

[6] Siqi Bao, Huang He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhen Guo, Zhibin Liu, Xinchao Xu. (2021). PLATO-2: Towards Building an Open-Domain Chatbot via Curriculum Learning. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (pp. 2513–2525). Online: Association for Computational Linguistics.

[7] Wang, W., Cai, X., Huang, C. H., Wang, H., Lu, H., Liu, X., & Peng, W. (2021). Emily: Developing An Emotion-affective Open-Domain Chatbot with Knowledge Graph-based Persona.

[8] Gibbeum Lee, Volker Hartmann, Jongho Park, Dimitris Papailiopoulos, Kangwook Lee. (2023). Prompted LLMs as Chatbot Modules for Long Open-domain Conversation. *Findings of the Association for Computational Linguistics: ACL 2023.* Association for Computational Linguistics. doi:10.18653/v1/2023.findings-acl.277

[9] H. Alkaissi, S.I. McFarlane. (2023). Artificial Hallucinations in ChatGPT: Implications in Scientific Writing. *Cureus*.

[10] Matthew Henderson, Ivan Vulić, Daniela Gerz, Iñigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrkšić, Pei-Hao Su. (2019). Training Neural Response Selection for Task-Oriented Dialogue Systems. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 5392–5404). Florence, Italy: Association for Computational Linguistics.

[11] Hui Su, Xiaoyu Shen, Zhou Xiao, Zheng Zhang, Ernie Chang, Cheng Zhang, Cheng Niu, Jie Zhou. (2020). MovieChats: Chat like Humans in a Closed Domain. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 6605–6619). Online: Association for Computational Linguistics.

[12] P. Muangkammuen, N. Intiruk, K. R. Saikaew. (2018). Automated Thai-FAQ Chatbot using RNN-LSTM. *2018 22nd International Computer Science and Engineering Conference (ICSEC)*, (pp. 1-4). Chiang Mai, Thailand.

[13] Guru Kiran Reddy K, Angad Pal, Shravan Krishna V, Rishi J, Saritha K. (2022). Cross Domain Answering FAQ Chatbot. *2022 International Conference on Advanced Computing Technologies and Applications (ICACTA)*, (pp. 1-4). doi:10.1109/ICACTA54488.2022.9752986

[14] Farhana Sethi. (2020). FAQ (Frequently Asked Questions) ChatBot for Conversation. *International Journal of Computer Sciences and Engineering, 8*(10).

[15] Wen Zhang, Heng Wang, Kaijun Ren, Junqiang Song. (2016). Chinese Sentence-Based Lexical Similarity Measure for Artificial Intelligence Chatbot. *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, (pp. 1-4). doi:10.1109/ECAI.2016.7861160

[16] Bayu Setiaji, Ferry Wahyu Wibowo. (2016). Chatbot Using a Knowledge in Database: Human-to-Machine Conversation Modeling. *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, (pp. 72-77). doi:10.1109/ISMS.2016.53

[17] Tarun Lalwani, Shashank Bhalotia, Ashish Pal, Vasundhara Rathod, Shreya Bisen. (2018). Implementation of a Chatbot System using AI and NLP. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST), 6*(3).

[18] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *North American Chapter of the Association for Computational Linguistics.*