University of Twente

Human Computer Interaction and Design EIT Digital Master School Interactive Technology University of Twente P.O. Box 217 7500 AE Enschede The Netherlands

M.Sc. Thesis

# ArgueBot: Enabling debates through a hybrid retrieval-generation-based chatbot

Iryna Kulatska

Supervisors from University of Twente Dr. M. Theune Prof. Dr. D.K.J. Heylen J.B. van Waterschoot, MSc

Supervisor from Findwise J. Bratt, MSc

2019

# Abstract

The goal of this study is to develop a debate platform, the ArgueBot, that is able to maintain a meaningful debate with the user for various topics. The goal of the chatbot is to carry out human-like debates with the users. The Arguebot uses a hybrid model, combining retrieval- and generative-based models. The retrieval model uses cosine similarity to compare the user input with the argument candidates for a specific debate. The generative model is used to compensate for the limitations of the retrieval model that is restricted to the arguments stored in the database. The Arguebot utilizes Dialogflow, Flask, spaCy, and Machine Learning technologies within its architecture. The user tests and the survey are used to evaluate the chatbot's performance. The user tests showed that there is potential in the Arguebot, but it needs better context understanding, a more accurate stance classifier and a better generative model.

# Acknowledgement

I would like to address big thanks for Mariët Theune, Jelte van Waterschoot and Jesper Bratt for being such rock stars in supervising this project. Without your feedback, support, and guidance, this thesis would not be possible. Thank you Dirk Heylen for the valuable feedback that helped me improve the final version of the thesis. Thank you Findwise, for providing me with an office space and the gallons of coffee. This project was very interesting to conduct, and I wish I had more time to improve it further. Finally, many thanks for my family, friends and Findwise colleagues for participating in the user tests and supporting me throughout the project.

# Contents

1	Intro	oduction 1
	1.1	Problem Statement
	1.2	Thesis Structure    3
2	Bac	kground and Related Work 4
	2.1	Argument mining
		2.1.1 Arguments and their components
		2.1.2 Stance classification
	2.2	Chatbots
		2.2.1 Types of chatbots
		2.2.2 Hybrid model
		2.2.3 Debate-chatbots
		2.2.4 Building a chatbot
		2.2.5 Evaluation
	2.3	Conclusion
3	First	t Implementation with Basic Functionalities 11
	3.1	Dataset 11
	3.2	Architecture
		3.2.1 Pre-Processing
		3.2.2 Model for data analysis
		3.2.3 Dialogflow
		3.2.4 Flask
	3.3	User tests and results
	3.4	Conclusion
4	Seco	ond Implementation with Machine Learning 25
	4.1	ArgueBot 2.0
		4.1.1 Dataset
		4.1.2 Architecture
	4.2	Stance classification with ML
		4.2.1 Data
		4.2.2 Methodology
		4.2.3 LSTM with Self-Attention Mechanism
	4.3	Generative Model
	110	4.3.1 Data 38
		4.3.2 Methodology
	4.4	Conclusion
_		
5	Fina	I evaluation of the ArgueBot 44
	5.1	Overview

	5.2	Survey	results	47						
		5.2.1	User Background	47						
		5.2.2	Debate information	47						
		5.2.3	Grammar	48						
		5.2.4	Conversation flow	48						
		5.2.5	Response quality	49						
	5.3	Conver	sation length	51						
	5.4	Conclu	sion	53						
6	Disc	ussion		54						
	6.1	Argue	Bot	54						
	6.2	Stance	Classification	56						
	6.3	Genera	itive Model	57						
	6.4	Hybrid	Model	57						
7	Cond	clusion		59						
Bil	Bibliography									
Fo	otnot	es		65						
Α	Appendix Survey ArgueBot 1.0									
В	Appendix Survey ArgueBot 2.0 72									

# Introduction

*Opinion is the medium between knowledge and ignorance.* 

— **Plato** ( c. 427 BC – c. 347 BC)

A debate can be defined as a "careful weighing of the reasons for or against something"<sup>1</sup>.

Debates can be tracked down to Ancient Greece, where philosophical minds were debating about politics and the nature of life. Throughout history, debating has been an essential tool in individual and collective decision making and has been helping in idea generation and policy building. Furthermore, the ability to articulate and evaluate arguments improves one's critical thinking and creativity (Keller et al., 2001).

In the time of flourishing social media worldwide, debates have become possible, where people with different backgrounds can engage in discussions about every possible topic across the globe. One such example is Doha Debates, that through live debates, videos, blogs, and podcasts evokes the discussions and collaborative solutions for today's global challenges such as global refugee crisis, Artificial Intelligence (AI), gender inequality and water shortage  $^2$ .

The latest advances in technology such as Natural Language and Speech Processing, Machine Learning algorithms, Argument Mining, Information Retrieval, and many others enabled human-computer interaction in the debate domain. One such example is the IBM Debater project, a conversational AI system that can give speech on a given topic and debate with humans <sup>3</sup>. The system uses several technologies: Argument Mining to identify argument components in the debate; Stance Classification and Sentiment Analysis to classify whether the argument is for or against a given topic; Deep Neural Nets (DNNs) and Weak Supervision, that is a Machine Learning

<sup>&</sup>lt;sup>1</sup>https://www.merriam-webster.com/thesaurus/debate

<sup>&</sup>lt;sup>2</sup>https://dohadebates.com/

<sup>&</sup>lt;sup>3</sup>https://www.research.ibm.com/artificial-intelligence/project-debater/

algorithm that improves the argument detection; and finally Text-to-Speech (TTS) Systems that convert text into spoken voice output and gives the Debater its voice.

In the meantime, chatbots are gaining more and more momentum as a new platform for human-computer interaction. According to Gartner, Inc by 2022, twenty-five percent of enterprises will have integrated virtual customer assistants and chatbots within their platforms <sup>4</sup>. However, current chatbot systems still have several limitations such as incorrect understanding of the context (meaning) of the user utterance, a lack of empathy and the inability to understand social and emotional cues that exist in human-to-human communication (Klopfenstein et al., 2017; Moore et al., 2017).

# 1.1 Problem Statement

The following research aims to create a chatbot that can maintain a meaningful debate with users on various topics. The goal of the chatbot, called ArgueBot, is to be able to carry out human-like debates with the users.

The problem statement for the following research is defined as:

How can a hybrid retrieval-generation-based chatbot maintain a debate with a user for various topics?

The problem statement can be divided into sub-questions:

- SQ:1 How can the model recognize and handle the arguments?
- SQ:2 How can stance classification be applied for the conversational agents?
- SQ:3 What is an appropriate model for the chatbot's response generation?
- SQ:4 How can human-like conversation with the chatbot be carried out in the debate domain?
- SQ:5 How can such a chatbot be evaluated?

The research presented in this thesis was carried out at Findwise AB, a consultancy company that provides search-driven solutions <sup>5</sup>. Findwise supported this project with guidance and testing.

<sup>&</sup>lt;sup>4</sup>https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technologytrends-for-2019/

<sup>&</sup>lt;sup>5</sup>https://findwise.com/en

# 1.2 Thesis Structure

### Chapter 2. Background and Related Work

This chapter elaborates on the background for the research topic and related work done within the field. Here, more information about existing methods for argumentation mining, building, and evaluating chatbots can be found. Moreover, research questions SQ: 1, 2, 3 ad 5 will be answered in relation to the previous work.

#### Chapter 3. First Implementation with Basic Functionalities

This chapter describes the chosen methods and user tests for the first implementation of the ArgueBot. Here research questions SQ: 1, 2, 3, 4 and 5 will be answered in relation to the first implementation of the ArgueBot.

#### Chapter 4. Second Implementation with Machine Learning

This chapter describes the changes made in the second implementation of the ArgueBot. Here, research questions SQ: 1, 2 and 3 will be answered in relation to the second implementation of the ArgueBot.

#### Chapter 5. Final evaluation of the ArgueBot

This chapter will present the results for the evaluation of the second implementation of the ArgueBot. Here, research question SQ: 5 will be answered in relation to the second implementation of the ArgueBot.

#### Chapter 6. Discussion

Here, the results presented in the previous chapter with their challenges and limitations will be discussed.

#### **Chapter 7. Conclusion**

This chapter will summarize the findings and propose how they can be further improved in future work. Here, all research questions will be answered with regard to the whole project.

# Background and Related Work

I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted.

— Alan Turing (Computing Machinery and Intelligence, 1950)

This chapter describes related work on Argument Mining, chatbots, their building, evaluation, and existing debate-chatbots done through literature review. The purpose of this chapter is to give an overview of the previous studies done on debate-chatbots and the related fields and describe what are the differences with the proposed study in this paper. This will make the basis for the chosen methodology for this study.

# 2.1 Argument mining

Argument (or argumentation) mining is a relatively new research field within the domain of Computational Argumentation that studies debate and reasoning processes by using artificial intelligence (Lippi and Torroni, 2016). The main goal of argument mining is to automatically extract arguments from the textual corpora (Lippi and Torroni, 2016) with their components and the relationships between these (Slonim and Aharonov, 2018), as well as analyze their stance, which is an overall position toward an idea, object or proposition (Somasundaran and Wiebe, 2010).

### 2.1.1 Arguments and their components

The ability to argue is a vital tool in supporting a specific claim or a standpoint, which is essential in debates (Kuhn, 1991). Walton (2009) describes an argument as a set of statements, made up of a conclusion, a set of premises, and an inference from the premises to the conclusion. Or as was explained in one of the Monty Pythons episodes "an argument is a connected series of statements intended to establish a proposition" (Chapman and Python, 1989).

Consider the following text extracted from the Wikipedia article "Ethics of artificial intelligence":

"Joseph Weizenbaum argued in 1976 that AI technology should not be used to replace people in positions that require respect and care, such as any of these: customer service representative [...], therapist [...], nursemaid for the elderly [...], soldier, judge, police officer. Weizenbaum explains that we require authentic feelings of empathy from people in these positions. If machines replace them, we will find ourselves alienated, devalued and frustrated. Artificial intelligence, if used in this way, represents a threat to human dignity [...]" <sup>1</sup>

Here, "AI technology should not be used to replace people in positions that require respect and care" is a **conclusion** (the claim, the core of the argument). "Weizenbaum explains that we require authentic feelings of empathy from people in these positions", "If machines replace them, we will find ourselves alienated, devalued and frustrated", "Artificial intelligence, if used in this way, represents a threat to human dignity" are the **premises** (statements that provide reason, evidence or support for the conclusion).

An **inference** is a process of drawing conclusions based on the premises and in the above-mentioned example would be:

- 1. humans need to feel empathy, that technologies cannot provide in the same way as professionals do;
- 2. the absence of empathy and authentic feelings can result in humans disappointment which threatens humans mental health;
- 3. therefore, AI should not replace professionals with positions that require respect and care.

Habernal and Gurevych (2017) proposed a model based on machine learning for identifying argument components containing feature sets: baseline lexical features; structural, morphological, and syntactic features; semantic, coreference, and discourse features; and embedding features. These sets of features were used to identify argument components and extract the arguments from the annotated forum posts.

Another method was proposed by Levy et al. (2017), who used it for detecting topic-relevant claims from the data extracted from Wikipedia. The study used claim sentence query to extract sentences with the word "**that**" followed by the claim topic, followed by any word from a pre-defined lexicon. This lexicon included words characteristic to the claims such as argue, disagree, argument, claim, conflict and others (Levy et al., 2017).

<sup>&</sup>lt;sup>1</sup>https://en.wikipedia.org/wiki/Ethics\_of\_artificial\_intelligence

Furthermore, information retrieval techniques can be used to structure the arguments (**indexing**), relate them to each other by computing how similar or dissimilar they are to each other, making it possible to find and **retrieve** the most relevant arguments and counterarguments (Stab et al., 2018; Ma et al., 2018; Wachsmuth et al., 2017; Wachsmuth et al., 2018; Zhu et al., 2018). Information retrieval can be defined as finding unstructured (does not have a clear, semantically distinguishable structure that is easy to understand by computer) data that complies with the information need from within a large collection of data (Manning, 2008).

### 2.1.2 Stance classification

Stance classification is a field within argument mining that helps to identify whether the argument is for or against the issue being debated. Mandya et al. (2016) proposed to extract the following features for stance classification: **topic-stance features** (specific words associated with topics); **stance bearing terminology** (words connected by adjectival modifier (amod) and noun compound modifier (nn) dependency relations that can indicate the stance in the argument); **logical point features** (extraction of words following the rule subject-verb-object (SVO) which might capture the claim); **unigrams and dependency features** (used to classify shorter posts).

Levy et al. (2017) proposed a method of claim stance classification in regard to a given topic. The study used precise semantic analysis of the debate topic and the claim (the sentiment of the claim towards its target), including target identification (through detecting the noun phrases in the claim), and contrast detection between the claim and the topic targets (through their relations), where each of these tasks had a separate machine learning classifier.

# 2.2 Chatbots

This section will present the recent developments of the conversational agents, also known as chatbots. Chatbot is a computer program that has an ability to mimic written or spoken human speech for interactions with humans (Kim et al., 2018).

### 2.2.1 Types of chatbots

Chatbots can be broadly classified into **generative** which generate a response based on natural language generation techniques (Kim et al., 2018; Le et al., 2018), and

**retrieval-based**, which select the most appropriate response by using information retrieval techniques (Zhu et al., 2018; Rakshit et al., 2019; Le et al., 2018).

**Retrieval-based models** require a database of possible responses to choose from. This model first retrieves the most possible candidates from the database that matches the current utterance and then selects the most appropriate response for the retrieval.

**Generative models**, by contrast, build responses simultaneously by using machine learning techniques. Here, the model is trained on a dataset consisting of real dialogues and is used to generate responses through "translating" inputs into responses. Statistical Machine Translation (SMT) models are some of the most recent models used for the generation of chatbot responses (Cahn, 2017).

### 2.2.2 Hybrid model

A hybrid model for chatbots that combined generative and retrieval models was previously explored in several studies. Tammewar et al. (2018) developed a personal assistant application for scheduling and cancelling the reminders. In this study, a graph-based retrieval model contained a set of nodes that represented different conversational states to navigate between and was used for the expected conversation flow. The generative model was applied when the conversation flow deviated from the expected.

Another study, Yang et al. (2019), proposed a hybrid neural conversational model by combining generation and retrieval models with a hybrid ranking module. The hybrid ranking module was used to select the best response from the generated and the retrieved candidates.

The model describes in this work is similar to the model proposed by Tammewar et al. (2018) as it also applies the same strategy for using the generated model when the retrieval model is not able to give a response, while the chatbots' purposes differ. The chatbot developed in this study is aimed to maintain a debate with the user instead of being a scheduling assistant as in the study done by Tammewar et al. (2018). It also differs from Yang et al. (2019) study, as it does not control the responses through a ranking module. This study prioritizes the retrieval module and applies the generative model to overcome the limitations of the datasets' limitations, while Yang et al. (2019) threat the responses from different modules as equally important.

7

### 2.2.3 Debate-chatbots

To date, at least two debate-chatbots were made: a chatbot Debbie, that uses a similarity algorithm to retrieve counter-arguments (Rakshit et al., 2019) and a chatbot Dave that used retrieval- and generative-based models separately (Le et al., 2018).

Chatbot Debbie used corpora compiled by (Swanson et al., 2015) containing controversial topics from the Internet Argument Corpus (Abbott et al., 2016) and dialogues from online debate forums. The authors were using the Argument Quality (AQ) regressor to choose the best arguments from the database containing statements for and against three controversial topics: death penalty, gay marriage, and gun control. Through Debbie chatbot, users were able to pick a topic and specify their stance (the chatbot assumes that the user utterance is always argumentative). The system then used a similarity algorithm based on the UMBC STS score (that combines lexical similarity features such as latent semantic word similarity and WordNet knowledge) to retrieve a ranked list of the most appropriate counter-arguments that was not previously used by the chatbot. The authors created clusters (groups of documents that are semantically similar (Manning, 2008)) with arguments to speed up the retrieval process. Chatbot Debbie continues the debate until the user terminates the chat. The chatbot was evaluated by comparing the average response times for different retrieval methods used for implementation (Rakshit et al., 2019).

Chatbot Dave (Le et al., 2018) also used Internet Argument Corpora (Abbott et al., 2016) for its knowledge base. The chatbot incorporates both a retrieval-based and a generative conversational model separately. The retrieval-based model used the Manhattan LSTM (MaLSTM) similarity model to learn the semantic similarity between messages and compare the user message with the knowledge base. To train and evaluate the MaLSTM model, a parallel corpus consisting of the Quora question pairs from Kaggle <sup>2</sup> was used. The Quora dataset was used as a "ground truth" for evaluation of the similarity model. Additionally, a context tracker function was implemented to keep track of the user and system responses. The generative model used a hierarchical recurrent (RNN) encoder-decoder architecture, where each word in the response was embedded using pre-trained word embeddings. The generative model was evaluated with a perplexity metric, distinct-1 and distinct-2 metrics (that is the number of distinct uni- and bi-grams in generated responses, scaled by the total number of tokens that are used to measure the degree of diversity of responses). These metrics were able to show the diversity of the generative model but were not useful for evaluating the conversational system. Instead, a rating system was

8

<sup>&</sup>lt;sup>2</sup>https://www.kaggle.com/c/quora-question-pairs

implemented in the chatbot interface, where the users were able to rate each chatbot responses from 1 (very bad) to 5 (very good) (Le et al., 2018).

The chatbot described in this work is different from the above-mentioned chatbots in several ways: firstly, the dataset that is used for the knowledge base of the chatbot is different resulting in different discussion topics within the chatbot; secondly, the model for implementation is different. While chatbot Debbie uses UMBC STS similarity score and chatbot Dave uses Manhattan LSTM similarity model, this project uses cosine similarity in the combination with the GloVe embedding vectors. Additionally, the final implementation of the chatbot presented in this work uses a hybrid model, combining both the retrieval and the generative models.

### 2.2.4 Building a chatbot

The Turing test that tests a machine's ability to perform intelligent behavior equivalent to human intelligence (Turing, 1950), inspired many researchers and engineers to develop multiple conversational systems. One such example is Eliza, a computer program that through pattern matching and specific phrasing could imitate humanto-human conversations (Weizenbaum, 1966). The most recent chatbot that passed the Turing test is Mitsuku (four-time Loebner Prize winner), built in Pandorabots <sup>3</sup> by using the artificial intelligence markup language (AIML). However, chatbots built using AIML have difficulties with maintaining a dialogue for a longer time (Shum et al., 2018) and are not able to extract complex information needed in the debate-domain.

Currently, there are many online tools available for building chatbots: Dialogflow, Microsoft Bot Framework (Cortana), IBM Watson Conversation, and many others. Among these, Dialogflow <sup>4</sup> is a free platform for creating interfaces based on natural language conversations which functionalities can be expanded by using webhooks (is a way to send information within different applications). Both Microsoft Bot Framework and IBM Watson Conversation have a free version that allows only a limited number of API calls per month.

### 2.2.5 Evaluation

When it comes to evaluating chatbot's performance, the most recent tool is ChatEval <sup>5</sup> that includes evaluation datasets with both human-annotated and automated baselines (Sedoc et al., 2018). The Turing test can be used to evaluate how human-

<sup>&</sup>lt;sup>3</sup>https://pandorabots.com/docs/

<sup>&</sup>lt;sup>4</sup>https://dialogflow.com/

<sup>&</sup>lt;sup>5</sup>https://chateval.org/

like the chatbot is. Chatbots can also be evaluated by conducting user tests and using surveys to determine user satisfaction (Higashinaka et al., 2018).

# 2.3 Conclusion

Concluding the literature review, there are various methods for the argument mining and development of the chatbots. Argument mining can include component extraction and stance classification. Chatbots can be retrieval-, generative-, or hybrid-based that include both model retrieval- and generative models. There are many online tools for building the chatbots, and some of these offer free versions. At least two debate-chatbots were previously made, retrieval-based chatbot Debbie and both retrieval and generative-based chatbot Dave.

In this chapter following research sub-questions were answered in relation to the previous work:

- SQ:1 *How can the model recognize and handle the arguments?* Argument extraction can be done through feature extraction and rule-matching("that" word) method. The arguments can either be retrieved or generated depending on the chatbot model.
- SQ:2 *How can stance classification be applied for the conversational agents?* Feature extraction, semantic and sentiment analysis, and machine learning can be used to classify the stance of the argument.
- SQ:3 *What is an appropriate model for the chatbot's response generation?* Depending on the chatbot's type, the responses can be produced by a retrieval, generative, or hybrid model. The retrieval model can use Manhattan LSTM or UMBC STS similarity score to extract the appropriate response, while the generative model can use hierarchical recurrent (RNN) encoder-decoder architecture. A hybrid model can use the generated model when the retrieval model is not able to give a response, or use a hybrid ranking module to select the best response from both retrieved and generated candidates.
- SQ:5 *How can such a chatbot be evaluated?* ChatEval tool, Turing test, and user tests in the combination with the surveys can be used for the chatbot evaluation.

# First Implementation with Basic Functionalities

The smart way to keep people passive and obedient is to strictly limit the spectrum of acceptable opinion, but allow very lively debate within that spectrum—even encourage the more critical and dissident views. That gives people the sense that there's free thinking going on, while all the time the presuppositions of the system are being reinforced by the limits put on the range of the debate.

> — Noam Chomsky (The Common Good (1998))

This chapter describes the first implementation of the ArgueBot platform, the design choices, and how it was tested. The goal of the first implementation was to build the base functionalities for interaction with the user. Henceforward, the ArgueBot chatbot will be referred to as an agent.

# 3.1 Dataset

The knowledge base for the chatbot consists of the ArguAna Counterargs corpus (Wachsmuth et al., 2018). Table 3.1 lists the 15 topics used in the dataset containing 1069 debates with 6779 points and 6753 counterpoints (see an example of how a debate is composed in figure 3.1) distributed between test, training and validation folders. Arguments consist of points with both pro and con stance towards the debate's statement. Each such point includes a conclusion, premises and an inference within its text, which are not separated or labelled (see chapter 2.1.1). Each debate has an introduction with the relevant information needed to make an argument. The data in the dataset was crawled from idebate.com <sup>1</sup>, an international debate education association for young people that offers debates written by experienced debaters from around the world. The ArguAna Counterargs corpus includes therefore

<sup>&</sup>lt;sup>1</sup>https://idebate.org/

high qualitative arguments, strengthened with citations. The downside of the corpus is its formal nature of argumentation, which might differ from the written arguments provided by the user in the chatbot. This corpus was chosen because of it including debate background and arguments with different stances, providing, therefore, stance labels for each argument and eliminating the problem of stance classification of the existing data.

Topic	Debates	Points	Counterpoints
Culture	46	278	278
Digital freedoms	48	341	341
Economy	95	590	588
Education	58	382	381
Environment	36	215	215
Free speech debat	e 43	274	273
Health	57	334	333
International	196	1315	1307
Law	116	732	730
Philosophy	50	320	320
Politics	155	982	978
Religion	30	179	179
Science	41	271	269
Society	75	436	431
Sport	23	130	130
Training set	644	4083	4065
Validation set	211	1290	1287
Test set	214	1406	1401
Total	1069	6779	6753

Tab. 3.1.: Distribution of debates, points, and counters over the topics in the dataset (Wachsmuth et al., 2018)

The first implementation used 12 debates marked as "Junior" from the dataset with the claims : "Ban online gambling", "Ban animal testing", "Kill One to Save Many", "Banning School Uniforms", "Poetry should not be taught in schools", "Raise the school leaving age to 18", "Ban the niqab and other face coverings in schools", " Dreaming of a white Christmas", "Introduce a "fat tax"", "Homework is a waste of time", "Every child should have a mobile phone", "Sponsoring children in developing countries". These debates were designed for the younger audience and included simplified topics with simplified arguments, which aligned with the purpose of the first implementation of creating the platform with some basic functionalities with the use of simplified debates. Each debate included at least six arguments (at least three arguments for and three against the main claim). Each argument included one point and one counterpoint. Each point and counterpoint were generally 4-8 sentences long each.



Fig. 3.1.: An example of a debate's architecture

An example of how the debates are composed can be seen in Figure 3.1. The debate has the main claim "Kill One to Save Many (Junior)" and consists of arguments and background information. This particular debate is unbalanced, consisting of three pro arguments and six con arguments. Each argument has a main claim and one point and one counterpoint. Each main claim is generally 3-10 words long which is often a conclusion of the argument's point. A counterpoint's goal is to argue against this main claim of the argument. For the "pro" argument a point has a "pro" stance and a counterpoint a "con" stance respectively. The "con" argument has an opposite architecture, a point here has a "con" stance, while a counterpoint has a "pro" stance. Note that every argument consists of a point and a counterpoint, but for readability, Figure 3.1 shows these only for "PRO 1" AND "CON 1" arguments and does not include any example of the background information.

# 3.2 Architecture

The overview of the architecture for the first implementation can be found in Figure 3.2. The data from the dataset is **pre-processed** in order to remove all the unnecessary information and saved into the database. **Flask** represents the ArgueBot platform that the user interacts with. **Dialoglow** is used to help understand the context of the user input. The **model** is used to create a response to the user. Flask connects the platform with the database, Dialogflow, and the model.

Every time the user chooses a new debate topic, the model finds the 100 most used words for that debate from the database and generates a debate object (memory object for the specific user to be used by the model) with response candidates that are also saved into the database. The 100 most used words in the dataset for that debate, hence called "debate-specific words", are then sent to the **argument entity** in Dialogflow through the API. When the user gives input in form of a chat message, the message is sent to Dialogflow that detects the intent (context) of that message using the debate-specific words in the **argument entity** and the sentence composition. If the user input is classified as an **argument**, it is then further analyzed by the model. The model checks how similar the user argument is to the argument candidates stored in the database and retrieves the appropriate response. If Dialogflow classifies the user input with some other intent, the model replies with a predefined response. Each section in Figure 3.2 marked with a blue rectangle will be described in more detail below.



Fig. 3.2.: Architecture of first implementation

### 3.2.1 Pre-Processing

Pre-processing included removing information within brackets, such as citations and explanations. Additional information for the debate backgrounds that explained the nature of the debate was also removed. These were removed by using regular expressions.

The debate names were changed through a written script from for example "This house Would Ban School Uniforms - Junior" to "banning school uniforms - Junior".

The "This House" wording format belongs to the British Parliamentary debate style that is a default format for many university societies. British Parliament consists of "Houses", thus "this house.." represents a motion to be discussed in the debate.

The name change included tokenizing the name, removing the first two tokens if they were "this" and "house", checking the tense of the verb and changing it to the present participle ("-ing") form. Tokenizing and verb-checking were implemented by using the spaCy library <sup>2</sup>. The arguments were then vectorized by using spaCy's GloVe vectors model package "en\_vectors\_web\_lg" and transformed into strings to save space. The use of these vectors will be further explained in the next section.

The pre-processed debates with their arguments were saved to SQLite database <sup>3</sup> to reduce the computing time for the model and make the retrieval process easier.

### 3.2.2 Model for data analysis

There are two main purposes for the model: one for handling the debate object (memory object to be used by the model) for each user and one to handle the analysis of the user input.

The **user-input handler** used the spaCy library to vectorize the input. It used cosine similarity to compare the vectorized user input to all the argument candidates for the chosen debate. It then retrieved the id of the argument that had the highest similarity and sent it to the debate model. The cosine similarity between two vectors is a measure that calculates the cosine of the angle between these vectors projected in a multi-dimensional space. Given two vectors  $\vec{a}$  and  $\vec{b}$ , their cosine similarity is

$$\cos\varphi = \frac{\overrightarrow{a} \cdot \overrightarrow{b}}{\|\overrightarrow{a}\| \times \|\overrightarrow{b}\|}$$
(3.1)

where  $\overrightarrow{a}$  and  $\overrightarrow{b}$  are multi-dimensional vectors over the term set  $T = \{t_1, \ldots, t_m\}$  and each dimension represents a word with its weight in the sentence. The cosine similarity is a non-negative number between 0 and 1 (Huang, 2008).

It then used NLTK Vader library <sup>4</sup> to classify the stance for the user input through sentiment analysis. The polarity of the user input (whether it has positive, neutral, or negative sentiment) was used to classify whether it was for or against the main claim of the debate. Positive sentiment was understood as a "pro" stance, negative sentiment as a "con" stance and neutral sentiment as undefined stance.

<sup>&</sup>lt;sup>2</sup>https://spacy.io/

<sup>&</sup>lt;sup>3</sup>https://www.sqlite.org/index.html

<sup>&</sup>lt;sup>4</sup>https://www.nltk.org/

The **debate-object handler** was managing multiple things:

- it randomly assigned the stance for the agent (for or against the main claim)
- it transformed vectors for all argument candidates from a string (you can read more about why the vector was saved as a string in the Data Pre-Processing section 3.2.1) to a Numpy vector <sup>5</sup>. Numpy is a Python library often used for computations. The model then rendered all the candidates with their id and the corresponding vector for the similarity computation done by the user-input handler.
- it assigned the argument with the highest similarity, received from the userinput handler, as an active argument for retrieval to the user. The retrieval process included comparing the user's stance (calculated with the help of polarity in the sentiment analysis in the user-input handler) and the agent's stance and retrieving the first sentence from the point or the counterpoint based on the agent's stance. If the user's stance and the agent's stance were the same, the agent would respond with "I agree" and when the stances differed with "I disagree". If the stance for the user input was undefined (when the polarity was neutral), the model would only retrieve the response without agreeing/disagreeing. If the agent's stance was "pro" the main claim of the debate and the active argument was "pro" as well, the model would retrieve the first sentence of the point in the argument; if the active argument was "con", i.e. against the main claim, the model would retrieve first two sentences from the counterpoint. The model then updated the argument by removing the used sentences from the database for the user. When the next user input was assigned to the already used argument (that had the highest cosine similarity), the next two sentences would be retrieved, until the argument became empty. If there were no sentences left to retrieve, the agent's response would be: "You already used this argument". This was done with the assumption that the user continues on the same argument as before because of the highest cosine similarity.
- it retrieved the 100 of most frequent words in the dataset for the debate (referred to as "debate-specific words") for the argument entity in the Dialogflow that helps with the argument detection (see section 3.2.3 for more information). It first tokenized all the sentences for all the arguments in the debate by using the spaCy library. Then it checked for each token if it wasn't a stop word (such as "the", "a", "an", "in" and other commonly used words that do not bear any necessary information) or a punctuation mark and saved the lemma form of the word, which is the base or the dictionary form of the word, to an array. It then used the Count function to retrieve the 100 most used words from the array.

<sup>&</sup>lt;sup>5</sup>https://www.numpy.org/

## 3.2.3 Dialogflow

Dialogflow is a platform for creating interfaces based on natural language conversations <sup>6</sup>. Dialogflow has a set of pre-built agents with **intents** that map user inputs to responses and **entities** that include information that can be extracted from the user input. Dialogflow has a console interface where it is possible to create intents, entities, fill in responses for specific phrases, and pre-train the intents with some phrases that the user would typically input. The Dialogflow API <sup>7</sup> makes it possible to access Dialogflow functionalities through the ArgueBot application and control the responses through a webhook. The webhook is a URL to the chatbot platform that sends the agent's response retrieved from the model back to Dialogflow.

The Dialogflow implementation included:

- argument entity with debate-specific words (the 100 of the most used words in the debate) that the model updated for every chosen debate. These entities helped to detect the argument intent and made intent-detection for multiple users possible;
- Default Welcome Intent, that recognized the greetings from the user;
- Default Fallback Intent, that when the other intents were not matched responded with *Try to start your argument with "I think..."*
- argument intent, that consisted of debate-specific words and helped to differentiate whether the user input was an argument or not (pre-trained on phrases: "there is test", "I think that test", "I argue that test", "in some test" where "test" was the default word in the argument entity and was included in the argument entity for every specific debate);
- stance intent for when the user was asking the agent for the stance (pre-trained on phrases: "Are you for or against the debate?", "What is your stance?", "Are you pro or con?" and such);
- why intent, for when the user misunderstood the agent or wanted to have more explanation on the specific argument (pre-trained on phrases: "why?", "what?", "I don't understand", "What do you mean by that?", "Can you explain more" and more)
- Small talk pre-built agent, that was customized through manual input to give specific responses for when the user used small talk phrases. Small talk could be manually customized for several areas: about agent (questions such as "Who are you?", "Are you real?", "You are bad" and others); courtesy (including phrases: "Great!", "Thank you!", "Well done" and so on); emotions (including

<sup>&</sup>lt;sup>6</sup>https://dialogflow.com/

<sup>&</sup>lt;sup>7</sup>https://github.com/googleapis/dialogflow-python-client-v2

phrases: "Ha ha ha!" and "Wow"), Hello/Goodbye (including greetings); about user (including phrases: "I'm bored", "I love you", "I'm waiting" and more); and other questions and phrases (including phrases: "You are wrong!", "Sorry" and others).

### 3.2.4 Flask

Flask is a micro web framework written in Python, that can be used to create web applications <sup>8</sup>. Ngrok cloud service <sup>9</sup> was used to create a tunnel to the local server on a randomly chosen sub-domain. This works in a similar way as a webhook, by providing a link to the local server running on the local machine that can be accessed by anyone and therefore can be used for testing. The drawback of the service is that the link is functional only when the local machine is on and connected to the internet, which limited the accessibility to the Arguebot platform.

Flask was used to create the ArgueBot platform and connect it to the database, Dialogflow, and the model. Flask Sqlalchemy<sup>10</sup> (an extension for Flask that adds support for storing objects into a SQL Database) was used to store user information such as session id, timestamp, conversation, and a debate model object for every chosen debate for every new user. This made simultaneous testing for different users at the same time possible.

Examine the following debate illustrated in figure 3.3 conducted by one of the participants on the topic "Raise the school leaving age to 18" during the user tests for the first implementation. Some of the user inputs Here "user" annotates the user input and "agent" annotates the ArgueBot response.

In the table 3.2, some of the user input/agent response pairs were picked up from the dialogue to illustrate the response model. Here, the table shows the intent of the user input and the response model chosen by the ArgueBot followed by the agent output as the resulted response.

<sup>&</sup>lt;sup>8</sup>https://palletsprojects.com/p/flask/

<sup>&</sup>lt;sup>9</sup>https://ngrok.com/

<sup>&</sup>lt;sup>10</sup>https://flask-sqlalchemy.palletsprojects.com/en/2.x/

USER: Hello

AGE	NT:	Let'	s sta	art t	he o	deba	te!	I	think	that	t foi	ced	ed	uca	tion	ac	hiev	/es
	lit	tle.	Wha	at do	o yo	u th	ink	ał	bout i	t?								

USER: It's not so much forcing I think

AGENT: Understood.

USER: I think the brain is still too much in development early on.

AGENT: Working at an early age can be an advantage in certain circumstances. Many families, particularly in countries with little welfare, need their children to bring income into the household.

- USER: I think the return on investment is solid. Children with more education can be more free in what they can do as an adult.
- AGENT: I agree! Working at an earlier age can help these families to survive. Furthermore, anyone who is having difficulties getting educational qualifications can gain an advantage by leaving school and gaining work experience.

USER: I agree with you that if you start working at a younger age, say 16, children can still learn later if they want to [*this user input is not understood by the model and results in triggering Default Fallback Intent*]

AGENT: Try to start your argument with "I think..."

USER: I think children can still learn when they're an adult

AGENT: If they are forced to stay in school then they will simply lose two years. The British government recognized this and introduced 21 000 extra apprenticeships in 2009.

USER: What is an apprenticeship?

AGENT: As well as these direct costs there is also a huge amount of losses that a country would face. Young people who leave school and enter the workforce contribute to the economy through taxes and contributions to pensions which the country would no longer receive if people remained in school.

USER: Thanks for talking!

- AGENT: You're welcome. Should we continue the debate? What is your next argument?
- USER: No, that's alright.

AGENT: Understood.

USER: Goodbye

AGENT: Bye! It was nice debating with you! Hope you enjoyed it as much!

Fig. 3.3.: A conversation conducted during the user tests for the first implementation of the ArgueBot on the debate topic "Raise the school leaving age to 18"

User Input	Intent	Response Model	Agent Output
Hello	Default Welcome Intent	response with a random ar- gument retrieved from the database	Let's start the debate! I think that forced educa- tion achieves little. What do you think about it?
It's not so much forcing I think	Small talk Di- alogflow agent	replies with customised re- sponse	Understood.
I think the brain is still too much in development early on.	argument intent	triggers argument intent, user-input analyser is then activated and determines that the polarity of the sentence is neutral and then retrieves first two sen- tences from the argument with the highest cosine similarity to the user input	Working at an early age can be an advantage in cer- tain circumstances. Many families, particularly in countries with little wel- fare, need their children to bring income into the household.
I think the return on invest- ment is solid. Children with more education can be more free in what they can do as an adult.	argument intent	Here the intent is classi- fied as an argument and the polarity is computed as "positive" which is un- derstood by the model as a "pro" stance	I agree! Working at an earlier age can help these families to survive. Fur- thermore, anyone who is having difficulties getting educational qualifications can gain an advantage by leaving school and gaining work experience.
I agree with you that if you start working at a younger age, say 16, children can still learn later if they want to	Default Fallback intent	this user input is not un- derstood by the model which triggers pre-defined Default Fallback response	Try to start your argument with "I think"
What is an apprenticeship?	argument intent	user asks for the clarifica- tion, but the input is classi- fied as an argument which triggers the argument in- tent with neutral polarity	As well as these direct costs there is also a huge amount of losses that a country would face. Young people who leave school and enter the workforce contribute to the econ- omy through taxes and contributions to pensions which the country would no longer receive if people remained in school.

 Tab. 3.2.:
 System's response model with some examples from the dialogue on the debate topic "Make voting compulsory"

# 3.3 User tests and results

This section will describe the evaluation strategy for the first implementation of the chatbot and its results. The evaluation was conducted through the user tests via a platform build for the testing purpose (see figure 3.4) and a survey created in Google Forms (see Appendix A). The purpose of these user tests was to identify how the users interact with the chatbot and how it can be further improved. Before conducting the user tests, the platform was tested by one colleague in order to assure the test quality. This test is not included in the results for the first implementation.

The user tests were conducted during three days and had 14 participants. The testers were mainly colleagues from Findwise that received the link to the ArgueBot through an internal communication system. Other testers were acquaintances contacted via Facebook. The user tests were anonymous and therefore there is no demographic information available for the participants.

The platform for the testing had two pages. The front page for the platform included information about the project, and the user consent form (see figure 3.4a). After the user gave his/her consent, the ArgueBot platform redirected the user to the main page (see figure 3.4b). The main page had the option box where the debate topic could be selected, information about the debate, the chatbot box for conversing with the Arguebot and the link to the survey. To make interaction anonymous, an identification code was given to the user and could be found on the top of the page. The user was later instructed to provide this identification code when filling in the survey in order to connect the survey answer to a specific conversation for further analysis.

The survey included:

- two yes/no questions that asked the users whether they found the background information clearly presented respective helpful
- three ranking questions where the users were asked to rank: on the scale from 1 to 10 how natural (human-like) the conversation flow with the chatbot felt; on the scale from 1 to 10 how satisfactory the grammar of the chatbot and its response quality was.
- the users were also asked to elaborate on their answers in an open-question form
- if the users wanted to leave additional feedback they had an opportunity to do so in a separate question at the end of the survey.

### Welcome to ArgueBot!

This research is a part of master thesis, entitled "ArgueBot: Enabling a debate through a multi-turn retrieval-based chatbot". The aim of the research is to create a chatbot that is able to maintain a meaningful debate with users on various topics throughout the whole interaction. The goal of the chatbot is to inform, challenge and provoke the user into critical thinking.

In this research, you will interact with the chatbot in the debate on different topics. After that, you will be asked to fill in the survey about your interaction.

The interaction with the chatbot will be saved for further analysis. Please, do not enter any private information into the chatbot such as your password or email.

- Your data collected by the chatbot:
- will not be passed on to third parties;
   will be treated confidentially;
   will only be used in anonymised form for this research;
   will be stored so that unauthorized persons cannot access it.

After you have given your consent, you will get an identification code. Please, use this code when filling in the survey later.

Be aware that the opinions stated by the chatbot do not represent its developer's opinions but are retrieved from the dataset containing multiple arguments with different points of view.

By checking in this checkbox, you confirm your consent, that you hereby understood your rights and agree to continue.

#### (a) The front page

Select a debate:	ArgueBot
Ban online gambling (Junior)	
Gambling is risking money on a chance outcome. If you are lucky you get back more money than you wagered. If you are unlucky you loss your stake. There are many different forms of gambling. Some are games of pure chance, such as lotteries, roulette or slot machines. Some involve some expertise or knowledge for example Javing card games such as noker or blackinke, to rebutine on the	Let's start the debate! In my opinion we cant enforce an online gambling ban.
outcome of a horse-race or football match. But in each case a lot of chance remains and there is a risk that the gambler will lose their money.	Governments can't actually do anything to enforce a ban on
Governments around the world have very different laws about gambling. In some cases it is banned completely, although criminal gangs often run illegal gambling. In some places, such as Macao, gambling is a well-known leisure	China does
industry with few limits on it. But in most countries governments allow some gambling but place strict rules upon how it operates. Debaters will need to find out what the laws are in their own country.	
The rise of online gambling on the internet since the 1990s has made covernment control much harder. In some countries like the USA, online	
gambling is against the law. But in many states online gambling is now legal. And companies can set up in any country where online gambling is allowed, and offer a service to interest users all over the world. So comblere service	Enter message

#### (b) The main page

Fig. 3.4.: The interface of the ArgueBot for the first implementation

All of the testers answered that the background information for the debate was clearly presented, while 10 from 13 even thought of it as helpful. The users that did not find it helpful either already knew about the debate topic or were not satisfied with the available topics.

The average ranking for the conversation flow with the chatbot was calculated to be  $5.5\pm2.1$ . Some users felt that having to start the sentence with "I think" (the Fallback message from the Dialogflow that the chatbot used when it could not interpret the intent of the user input) felt unnatural and discouraged them from engaging more in the debate discussion. Some users commented that "You already used this argument" response from the chatbot (see section 3.2.2) broke the continuity of the conversation and was often wrong.

The average ranking for the response grammar was  $8 \pm 1.5$ . Here, the testers mostly commented on the missing spaces between the words, which was either the result of pre-pocessing or the inherited mistakes from the dataset. The average ranking for the response quality was  $5.8 \pm 2.1$ . Some testers felt that the chatbot responses were irrelevant to their input as if the chatbot failed to understand the context of their argument. The "I agree"/"I disagree" statement used by the chatbot was according to the majority of the testers (8 from 13) disconnected from the user input and in many cases wrong.

# 3.4 Conclusion

The first implementation of the debate platform included the basic functionalities such as chatting on the simplified debates, argument retrieval, dynamic entities, stance classification through sentiment analysis and simultaneous use of the platform by different users at the same time. The user tests showed that sentiment analysis for stance classification is not sufficient and should be changed to another method, forcing the user to start the argument with "I think" as a Fallback message and "You already used this argument" felt unnatural. In the next chapter, the second implementation of the ArgueBot will be presented with changes made based on the feedback from the user tests in the first implementation.

This chapter answered following research questions:

SQ:1 *How can the model recognize and handle the arguments?* Dialogflow was used to recognise the context (meaning) of user inputs.

- SQ:2 *How can stance classification be applied for the conversational agents*? In the first implementation polarity (whether the sentence is positive, negative or neutral) from the sentiment analysis was used for the stance classification.
- SQ:3 *What is an appropriate model for the chatbot's response generation?* If the argument intent was triggered, the input was further handled by the model for the retrieval of the argument with the main claim that had the highest similarity to the user input.
- SQ:4 *How can human-like conversation with the chatbot be carried out in the debate domain?* The first implementation used Small talk pre-built agent available in Dialogflow, argument retrieval through cosine similarity, stance classification to improve the argument responses and make the conversation more human-like. The Default Fallback message meant to lead the conversation back to the debate if it deviated from the intended flow.
- SQ:5 *How can such a chatbot be evaluated*? The platform was built to perform the user tests in the combination with the survey that collected user's feedback on their interaction with the chatbot and its performance.

# 4

# Second Implementation with Machine Learning

[Language] makes infinite use of finite means

— Wilhelm von Humboldt

This chapter presents the second and final implementation of the ArgueBot platform and the design choices. The user test section for this implementation is more extensive and will, therefore, be described in the separate chapter 5.

# 4.1 ArgueBot 2.0

This section will introduce the changes made in the second implementation of the ArgueBot and their motivations. The major changes, such as stance classification and the generative model, will be described in a separate section each.

## 4.1.1 Dataset

The dataset for argument retrieval was extended from the Junior topics to debates in the test dataset in the ArguAna Counterargs corpus (Wachsmuth et al., 2018). Some debates belonged to different topics which resulted in some duplicates. The code for filtering out the duplicates was included in the pre-processing code and resulted in 175 debates saved to the database from 214 existing in the dataset. The distribution of debates in the second implementation with their points and counterpoints for each topic can be found in Table 4.1.

# 4.1.2 Architecture

Figure 4.1 shows the architecture for the second implementation. Similar to the first implementation, the dataset is first pre-processed and then saved into the database. Here, junior debates were replaced with more complex debates (see the distribution for different topics for selected debates in Table 4.1). Because of the higher argument complexity, more extended pre-processing was applied. This included removing

Торіс	Debates	Points	Counterpoints
Culture	7	54	54
Digital freedoms	9	61	61
Economy	17	125	125
Education	10	76	76
Environment	5	36	36
Free speech debate	9	58	58
Health	10	77	77
International	30	233	233
Law	19	134	134
Philosophy	10	85	85
Politics	26	194	194
Religion	5	36	36
Science	8	57	57
Society	6	39	39
Sport	4	30	30
Total	175	1295	1295

Tab. 4.1.: Distribution of debates, points, and counters over the topics in the database for the second implementation

notes, annotations, references, and footnotes. Moreover, the number of the most used words for extracting and updating the argument entity in the Dialogflow was increased from 100 to 300.

The interaction with the user is quite similar to the first implementation, but here, instead of using sentiment analysis, a stance classifier developed using Machine Learning (ML) was added. Moreover, instead of the Fallback message *Try to start your argument with "I think..."* that was used by the Fallback intent (when the Dialogflow could not match the user input to any of the available intents), the user would get a generated argument created by the Generative Model.

The model that retrieved the sentences from the database, instead of choosing one argument with the highest similarity to the user input, chose two arguments instead. If the first argument did not have any sentences left to retrieve, the next available sentence from the second argument was retrieved. This was done instead of replying with "You already used this argument", which according to the user tests of the first implementation was often wrong and felt unnatural. Moreover, instead av retrieving two sentences at the time from the argument as was done in the first implementation, one sentence was retrieved instead. This was done to maximize the number of candidates for the retrieval.

To differentiate between different models used in this implementation, the model used to retrieve a response for the specific user from the database, will be hence referred to as the Retrieval Model. To differentiate responses created by different models during the user tests, the responses created by the generative model included "GR" token at the end of the sentence. The chatbot's model is hence hybrid, as it uses both retrieval and generative models. Additionally, a new intent was created to explain the purpose of "GR" at the end, if the user would ask what "GR" means.

The major changes made in the ArgueBot are marked in figure 4.1 with dashed rectangles. They will be explained thoroughly in next two sections : section 4.2 for the Stance Classification with ML and section 4.3 for the Generative Model.



Fig. 4.1.: Architecture of second implementation

Examine figure 4.2 showing a part of the debate conducted in by one of the participants on the topic "Making voting compulsory" during the user tests for the second implementation. Here, "user" annotates the user input, "agent" annotates the Argue-Bot response and the response flow is explained within the brackets. The agent's stance is for ("pro") making voting compulsory. A small part of the original dialogue was removed as it was most probably a typo made by the user at the beginning of the conversation.

In the table 4.2, some of the user input/ agent response pairs were picked up from the dialogue to illustrate the response model. Here, the table shows the intent of the user input and the response model chosen by the ArgueBot followed by the agent output as the resulted response.

- USER: I think voting should not be forced. instead people should be educated on its importance
- AGENT: I disagree! A benefit of compulsory voting is that it makes it more difficult for special interest groups to vote themselves into power.
- USER: I believe that people forced to vote are more likely to vote randomly, even stupid parties
- AGENT: I disagree! The benefits obtained from compulsory voting cannot be gained from any of the strategies mentioned by the opposition.

USER: what benefits?

- AGENT: Compulsory voting can enhance a sense of community, as everyone is in it together.
- USER: yeah but that is why government should foster civic education for all citizens with the right to vote

AGENT: I disagree! A notable example would be the disproportionate influence of agriculture in policy making as seen in both European politics and well as American with enormous amounts of subsidies for farmers who represent a minute percentage of the population.

USER: democracy is imperfect but at least give everyone the right to express his idea. Everyone has its own interest, no one is a farmer

- AGENT: if people wish to consider their own safety then why should the state not to be able to decide what their own people are GR
- USER: I said that this is part of a democratic system. no one has the same interests but at least everyone has the right to express his ideas
- AGENT: I disagree! The outcome of the election therefore reflects less the will of the people.
- USER: yes but I prefer a fair and transparent system such as a democracy rather than a oligarchy in which few people decides what is good for everyone
- AGENT: I agree! The right to vote in a democracy has been fought for throughout modern history.
- USER: thanks <3

AGENT: You're welcome. Should we continue the debate? What is your next argument?

Fig. 4.2.: A conversation conducted during the user tests for the second implementation of the ArgueBot on the debate topic "Making voting compulsory"

User Input	Intent	Response Model	Agent Output
I think voting	argument	stance classification deter-	I disagree! A benefit of
should not be forced. instead	intent	mines "con" stance, the first sentence from the	compulsory voting is that it makes it more difficult
people should		point from the argument	for special interest groups
be educated on		with the highest similarity	to vote themselves into
its importance		0.92 for the argument with	power.
		the claim "It will reduce	
		the power of special inter-	
		est groups" is returned	
democracy is im-	Default	triggers generative model	if people wish to consider
perfect but at	Fallback	that ends the sentence	their own safety then why
least give every-	intent	with "GR" token for eas-	should the state not to be
one the right to		ier differentiation between	able to decide what their
express his idea.		generative and retrieval re-	own people are GR
Everyone has its		sponse models	
own interest, no			
one is a farmer	lleveleerll in	not minutes the ment of the labor	Compulsory yeting con on
what benefits?	wily in-	retrieves the next available	bance a sense of commu
	tent	argument "There are alter	nitu as overvene is in it to
		nativos that tackle the real	acthor
		causes of voter disengage.	geniei.
		ment" that had the highest	
		similarity in the previous	
		turn	
thanks <3	Small	pre-defined response from	You're welcome. Should
	Talk	Dialogflow	we continue the debate?
	agent		What is your next argu-
			ment?

 Tab. 4.2.:
 System's response model with some examples from the dialogue on the debate topic "Make voting compulsory"

# 4.2 Stance classification with ML

Stance classification of the user input (determining whether the user is in favor of or against the main claim) for the second implementation was developed by using Machine Learning technologies instead of using sentiment analysis. Six different binary Machine Learning methods for classification were tested in two experiments. Binary classification is one of the most common tasks within the machine learning domain and is commonly used for classification when only two classes are present. The model that showed the best performance during the two experiments were then used by the chatbot in the user tests. This section will first present the dataset used for the classification, the methodology of the experiments done, and finally, explain the chosen classifier more in-depth.

29

### 4.2.1 Data

To create the dataset for the stance classifier, the ArguAna Counterargs corpus (Wachsmuth et al., 2018) was used. This included more thorough pre-processing:

- all the references, footnotes, additional information within brackets were removed from arguments by using regular expressions.
- the sentences inside every point and counterpoint were then split by using the spaCy library <sup>1</sup>.
- a csv file was then created by saving each sentence and a corresponding stance from the dataset in a binary form (1 for "pro" and 0 for "con").
- the csv file was then manually reviewed to remove the references not captured by regular expressions.
- the additional dataset for stance classification "IBM Debater Claim Stance Dataset" (Bar-Haim et al., 2017) was added to the existing csv file to improve the classifier, which resulted in a file with 49544 lines, where every line included a sentence with a respective stance as a label.
- the resulting file was then split into train/validation/test files with ratio 75/15/15. The model is used to fit the parameters for classification on the training dataset (the model learns the features of the input and their relation to the corresponding stance). The validation dataset can be used to compare the performance of the model during the training and tune the hyper-parameters used by the model (it treats the input data in the validation dataset as unseen, predicts their stance and evaluates how many of these predictions are correct). The test dataset is used to provide an unbiased evaluation of how well the final model is fit on the unseen input data from the test dataset (it is similar to the validation process, but happens after the training is done). During the experiment, the validation set was only used for tuning the best performing classification model because of the time constraints. Table 4.3 shows the distribution for different stances in the dataset, where the sentences with the pro stance have 3% higher distribution, therefore, the dataset is slightly imbalanced.

	Training	Validation	Testing
Data set	34686	7432	7432
Pro stance	18254	3907	3938
Con stance	16432	3525	3494

Tab. 4.3.: The Dataset used for the Stance Classification in number of l	ines
--	------

<sup>&</sup>lt;sup>1</sup>https://spacy.io/

## 4.2.2 Methodology

This section will provide an overview of the experiment done to determine the most suitable Machine Learning classification model for the given data. A general overview of different ML models used will be described here, while the chosen model will be explained in more details in the next section.

Google Collaboration (Colab) is a free Jupyter notebook environment that allows one to use a limited amount of a hardware accelerator such as GPU or TPU. While TPU is working exclusively with Tensorflow (an open source machine learning library developed by Google), the GPU can be used with various machine learning libraries. The reason why hardware accelerators are preferable is because of their computational speed, as they can train models way faster than with the CPU available on modern computers.

A set of experiments were, therefore, conducted on Google Colab to test different machine learning models for stance classification. The PyTorch machine learning library for Python  $^2$  was used for six different machine learning models: CNN, Self-Attention Networks, LSTM, LSTM with self-attention mechanism, RCNN, and RNN networks. The code was based on the existing solution provided in GitHub  $^3$ .

CNN (convolutional neural networks) is a commonly used method for image classification, that uses kernels (often 2x2, 3x3, 4x4 pixel squares) to first select important features and save them in hidden layers and then run through the new images in order to classify them with the help of these features. For NLP tasks the method works in the same way but by using tokens instead of pixels. The CNN model usually works well with data that has the same size (such as images with the same resolution) as it is able to extract the characteristic features for classification purposes.

The Self-Attention Networks (SAN) uses weights distributed for different words in the sequence to find which combinations of the words are the most important. The method was successfully used for a range of NLP tasks, such as machine translation, sequence labeling, relation extraction, and others.

RNN (recurrent neural networks) are networks that have loops in them. When dealing with text, RNNs understand words they encounter later in the sentence given the words they have encountered earlier. As the distance between the words grows, they cannot make the correct connections and therefore, focus more on the words close to one another.

<sup>&</sup>lt;sup>2</sup>https://pytorch.org/

<sup>&</sup>lt;sup>3</sup>https://github.com/prakashpandey9/Text-Classification-Pytorch
LSTM stands for Long Short-Term Memory and is built to overcome the beforementioned issue in RNNs. LSTMs are capable of learning long-term dependencies with memory cells that can maintain their state over time through control gates, that are controlling which information should be let in or out.

LSTM with a self-attention mechanism (LSTM SAM) enables RNNs to learn the correlation between the current words and the previous part of the sentence and save them into memory cells.

RCNN (Recurrent Convolutional Neural Networks) captures contextual information with the recurrent structure and constructs the representation of text using a convolutional neural network.

The first experiment included training each model 5 times with 10 epochs (number of iterations that the machine learning model uses to go through the same data) and choosing the model with the best accuracy. The hyper-parameters used in this experiment were: 32 batches (number of samples that will be propagated through the network) and 512 hidden features (number of features that the network learns from each sample). The accuracy is the number of correct predictions divided by the number of all samples (Müller and Guido, 2016). For the simplicity and because of the time constraints the same hyper-parameters were used for all the models.

Table 4.4 shows the accuracy for the different models for the first experiment. Comparing the accuracy for the training and the validation datasets can help developers to identify if the model is overfitting (if the validation accuracy is lower than the training accuracy) or underfitting (the opposite) during the training. When the model is overfitted, it works well on the training set (it learned rules specifically for the training set) but is not able to generalize to new data. Underfitting occurs when the model is not able to capture all the aspects of the training data (Müller and Guido, 2016). From table 4.4, we can observe that the SAN and LSTM SAM are prone to overfitting, CNN and LSTM are prone to overfit slightly, while RNN is prone to underfit. To overcome these issues the model can be tuned through adjusting the hyper-parameters so that they satisfy the trade-off of over/under-fitting. Due to the time constraints, the tuning was done if needed only for the best performing model.

The testing accuracy shows the accuracy of the trained model on the test dataset, namely on the data the model has not seen before. This metric is therefore used to measure the performance of the trained model.

Even though CNN showed the best test accuracy, the model had restrictions towards the input size and was therefore not suitable for classification of sentences with different size. If the user input length was smaller, say three words, the model would not be able to compute the predictions. LSTM did not have any restrictions on the input size and showed a testing accuracy that was just a fraction smaller than CNN (0.31%). LSTM model was therefore chosen as a stance classifier for the chatbot. Because the model was prone to overfit just slightly, the decision was made to not tune it further but to keep the same hyper-parameters.

Method	Training	Validation	Testing
SAN	98.27	61.87	63.71
CNN	69.95	68.00	69.12
LSTM	68.65	68.20	68.81
LSTM SAM	97.64	61.48	62.26
RCNN	96.91	61.42	63.04
RNN	66.31	68.18	68.68
Data set	34686	7432	7432

Tab. 4.4.: Comparison between different machine learning models for stance classification

Sometimes looking only at accuracy is not enough to measure the performance of the model. The dataset can be imbalanced with one of two classes much frequent than the other one. This can result in the model making false positives (incorrect positive predictions) and false negatives (incorrect negative predictions) that accuracy measurement does not take into consideration. F-score is a harmonic mean of two other performance metrics, precision and recall, and can provide a more realistic measure of a test's performance by using both of these metrics. Precision shows how many samples predicted as positives are indeed positive, while recall measures how many of positive samples are captured by the model (Müller and Guido, 2016).

The general formula for F-score is the following:

$$f\beta - score = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall},$$
(4.1)

where  $\beta$  is a positive real that allows to control the importance of precision and recall. The F-score is evenly balanced when  $\beta = 1$ , it favours precision when  $\beta > 1$ , and recall otherwise (Sokolova et al., 2006). The classifiers with low F1-score, can indicate that there is a problem with false positives or false negatives but can't specify it. F2-score ( $\beta$  is equal to 2) weights recall higher than precision and is used where it's more important to classify correctly as many positive samples as possible, rather than maximizing the number of correct classifications.

The second experiment included measuring F1- and F2- scores as well as testing six text inputs with different meaning (the ideal model would have captured the differences between these sentences): "Terrorism is wrong", "Terrorism is not wrong", "Terrorism is good", "Terrorism is very good", "Murder is good", "Murder is bad". The results from the second experiment can be found in the table 4.5. The table shows accuracy, F1-score (where  $\beta = 1$ ), F2-score (where  $\beta = 2$ ) and the stances for the text where "c" stands for "con", "p" for "pro" and "n" for "none" with the ground truth "pccccp" or "cppppc". The reason behind two ground truths is because the chosen sentences were not based on any debate topic available in the dataset and therefore it is hard to determine if they are "pro" or "con". The main goal of choosing these specific sentences. In retrospect, if the sentences would be more related to a specific topic in the dataset, they could have helped to compare the different ML models more conveniently.

To illustrate the difference between test accuracy metric and F-scores, one can look at the results for LTSM, where test accuracy showed almost 69% while F-scores showed 0%. Here, 0% F-scores indicate a problem with recall or precision, while the input test shows all the inputs being classified as "con" stance. We can, therefore, assume that the LSTM model classified all the values in the test set as a "con" stance resulting in many false negatives.

The second experiment was done in the middle of the user test, as it became apparent that the LSTM stance classifier was not performing well enough despite its high test accuracy. The same models from the first experiment that were previously saved to the files were used again in the second experiment. The effects of this change on the user tests will be further discussed in section 5.3.

ML Model	Test Accuracy	F1-score	F2-score	Input (correct)
SAN	63.71	58.00	77.01	ppppcp (3 or 3)
CNN	69.12	07.12	15.39	nnnnnn (0)
LSTM	68.81	00.00	00.00	cccccc (3 or 2)
LSTM SAM	62.26	64.76	81.63	cpcpcp (3 or 3)
RCNN	63.04	55.01	74.80	ppcccc (4 or 2)
RNN	68.68	04.04	09.18	cpcppp (2 or 4)
Data set	34686	7432	7432	

 

 Tab. 4.5.: Comparison between different machine learning models for stance classification in the second experiment, where "c" stands for "con", "p" for "pro" and "n" for "none"

As can be seen from Table 4.5, the CNN, LSTM, and RNN had the highest testing accuracy, but failed in the F1- and F2- scores and the input test. LSTM SAM had the highest F1- and F2- scores and performed on the input test and was therefore

chosen as a stance classifier for the chatbot. Due to overfitting problem discovered in the first experiment, the model was further tuned by increasing the batch size and the number of features for learning which thus improved LTMS SAM model's generalization (with training accuracy 78.93%, validation accuracy 62.64%, testing accuracy 62.12%, F1-score: 59.64% and F2-score: 79.44%). This model used hyper-parameters: 64 batch size and 768 hidden features.

To illustrate the difference between interaction with the chatbot using different stance classifiers two dialogues were chosen for the same debate "Banning the development of genetically modified organisms" from the user tests and are shown in figure 4.3. Figure 4.3a shows the ArgueBot version using the LSTM as a stance classifier, while figure 4.3b shows the dialog using the LSTM SAM as a stance classifier. The blue color to the left denotes the user input, while the green color to the right denotes the chatbot response.

When analyzing the dialog in figure 4.3a, two problems can be recognized: incorrect stance classification (the chatbot agrees with each of the user's arguments), an unrelated response (the user is talking about the danger of genetically modified organism, and the chatbot responds with the benefits of genetically modified food). The combination of these problems creates a disruption in the conversation, and the user might feel that the chatbot completely misunderstands him or her, as the chatbot continues to agree with everything that the user says. Moreover, the user in this example seems to be testing the system by trying the sentences with the semantic differences ("Genetic modification is good/bad"). This inspired to include such sentences when conducting the experiment for the classifier, mentioned previously.

The second dialog in figure 4.3b uses the LSTM with the self-attention mechanism as the stance classifier. It is able to classify the user inputs with different stances (it responds with both "I agree" and "I disagree" statements), but the chatbot is still not able to analyze the more complex user inputs, such as questions. The chatbot was designed with the assumption that the user will input arguments on the topic and while it can handle simple questions for clarification such as "why?" or "what?", the question "Agree with what?" is too complex to handle. Here, a clear example of the dataset limitations can be observed: the agent response "*There are two problems associated with scientifically testing the impact of genetically modifying food*" denotes the beginning of the argument and results in an obvious response from the user that asks "*What problems?*", that the agent assumes is an argument and responds with a new argument from the dataset. Moreover, the user in this example got the generated response that ends with "GR", that while grammatically correct, is not related to the debate topic of genetically modified food.



(a) LSTM stance classifier



- (b) LSTM SAM stance classifier
- **Fig. 4.3.:** Comparison between two dialogues using different classifiers for the same debate topic "Banning the development of genetically modified organisms", where the blue color represents the user and the green color represents the agent

#### 4.2.3 LSTM with Self-Attention Mechanism

The LSTM SAM was chosen as a classification model for the ArgueBot based on the experiments demonstrated in the previous section and will be further explained here. First, the overview of the LSTM networks will be given, then the LSTM SAM will be explained with the help of figure 4.4.

The LSTM method was first proposed by Hochreiter and Schmidhuber (1997) to overcome the vanishing/exploding gradient problem in the RNN networks. This problem occurs when the gradients that carry the information needed for updating the weights and setting up the network become too small/big and result in the model not being able to learn. The LSTM networks use a gating mechanism that controls the degree to which the LSTM units keep the previous state and store the extracted features in them.

Figure 4.4 shows the architecture for the LSTM networks with the self-attention mechanism (LSTM SAM). First, the words are transformed into their vector representations " $v_n$ " (300-dimensional GloVe embeddings), which are then fed into the LSTM embedding layer. "A" here represents a chunk of the LSTM network that has a chain-like connected structure, where every repeating chunk "A" has three gates that control the information flow for its memory cell and pass it to the hidden state " $h_n$ " containing the word features and to the next chunk "A" in the chain. The three gates are: the input gate that regulates how much of the new information the cell should keep, the forget gate that regulates how much of the existing information the cell should throw away or keep, and the output gate that regulates what information to pass to the next chunk in the network and the hidden state. The attention layer then finds the contribution of each word to the whole input by assigning weights " $w_n$ " to each word. The sentence embeddings "M" is computed as the sum of these weights in the vector matrix, where each vector represents an aspect, or component of the semantics (long sentences can have multiple components) belonging to a "pro" and a "con" class. The sentence embedding in the attention mechanism is able to provide the semantic representations of the input (long term dependencies), allowing LSTM to carry only shorter term context information around each word (short term dependencies) and in doing so relieves some of the memory load from the LSTM network (Lin et al., 2017). The output "r" is a sentence feature vector, containing the sentence embedding for "pro" and "con" class. When testing a new, unknown input (such as the user input in the chatbot), the model returns the class with the highest weight.



Fig. 4.4.: Architecture of the LSTM with Self-Attention Mechanism

## 4.3 Generative Model

The generative model was added to the ArgueBot to compensate for the limitations of the retrieval model that is restricted to the arguments stored in the dataset. In the second implementation, this model was added instead of the chatbot responding with the "You already used this argument" Fallback message. The ArgueBot 2.0 is, therefore, using a hybrid model, a combination of retrieval and generative models. In this section, the dataset used for training of the generative model, the methodology and the experiment will be presented.

#### 4.3.1 Data

The ArguAna Counterargs corpus (Wachsmuth et al., 2018) served as the dataset for the generative model. The corpus was first pre-processed as was done for the stance classifier, where the references and footnotes and brackets were removed with regular expressions. The sentences in every argument were then split by using the spaCy library <sup>4</sup>. The sentence pairs were then saved into a csv file. For example, if the argument consisted of 5 sentences, the first and the second, the second and the third and so on until the fourth and the fifth sentence, would be the input/output pairs in the csv file. Then, the resulting csv file was manually reviewed to remove the lines including references that Regex was not able to capture.

<sup>&</sup>lt;sup>4</sup>https://spacy.io/

As mentioned before, the ArguAna Counterargs corpora consist of high-quality arguments, but these arguments have a more formal nature of argumentation which differ from the conversational style of debating. The dataset used in (Carstens and Toni, 2017) was therefore added to the existing file to enrich it with conversational data. The dataset in (Carstens and Toni, 2017) was composed of the Internet Argument Corpus (IAC) (Walker et al., 2012) (comprised of text extracted from the debating website 4forums), the AIFdb corpus (Lawrence and Reed, 2014) (comprised of discussions that have taken place on the BBC Moral Maze and argumentative microtexts) and the News articles corpus (Carstens and Toni, 2015) (comprised of collection of publications concerned with the UK Independence Party (UKIP)).

An effort was made to add a more recent Internet Argument Corpus (IAC) <sup>5</sup> to the composed dataset which resulted in the csv file being too large to process. The decision was then made only to use the dataset containing sentence pairs from the ArguAna Counterargs corpus (Wachsmuth et al., 2018) and the dataset used in (Carstens and Toni, 2017) (without the UKIP news articles corpus), for training the generative model resulting in 47322 input and output pairs of sentences.

#### 4.3.2 Methodology

This section will describe the methodology and tools used for implementing the generative model. It will also describe the experiment conducted to determine which Luong score function to use for generating responses.

Similarly to the stance classifier with ML, Google Colab was used to pre-train the model on their GPU for much faster training. A Pytorch sequence-to-sequence (seg2seq) generative model for chatbots was used for generation of new arguments<sup>6</sup>. Its model is using a movie dialog corpora with sentence pairs, that was replaced with the created csv dataset instead. The code from the tutorial was adjusted to fit the requirements of ArgueBot, as the sentences in the created dataset were longer and more complex than that of the movie dialogs.

To prepare data for the model, the sentences were first normalized by filtering out the sentences longer than 50 words, which resulted in 45352 pairs of sentences. All the symbols that were not words were removed and the sentences were tokenized. The resulting tokens were added to the dictionary that consisted of 25511 words. The words in the sentences then were converted to their corresponding indexes in the dictionary and fed into the model.

<sup>&</sup>lt;sup>5</sup>https://nlds.soe.ucsc.edu/iac2

<sup>&</sup>lt;sup>6</sup>https://pytorch.org/tutorials/beginner/chatbot\_tutorial.html

Seq2seq models take an input sequence and return an output sequence using fixedsize encoder-decoder architectures and can be used for Machine Translation, Text Summarization, Conversational Modeling, Image Captioning, and more. Both encoder and decoder use two separate GRU recurrent neural networks. The GRU is a multi-layered Gated Recurrent Unit (Cho et al., 2014), that similarly to the LSTM networks eliminates the vanishing/exploding gradient in the RNNs through the gate architecture. While the LSTM networks use three gates (input, forget and output), the GRU has only two gates (reset and update). The reset gate controls how much of existing information to forget between the chunks in the network, while the update gate controls what information to add to existing information and what information to throw away (a combination of input and forget gates in the LSTM). The GRUs don't have any cell states as LSTMs do and use only hidden states to transfer the information and therefore require less computational power than the LSTM networks.

Figure 4.5 illustrates the architecture of the seq2seq encoder-decoder mechanism with GRU and attention layers. "<SOS>" and "<EOS>" tokens represent the start and the end of the sentence and are added to every input fed into the model. Vectors " $v_n$ " are the words' indexes in the dictionary.

The encoder GRU iterates through every token in the input sentence and outputs the output vector " $w_n$ " and a hidden state vector " $h_n$ ". It uses bi-derectional GRU that has two independent RNNs, one for feeding the input in the normal sequential order represented by chain-like network chunks "A" and one for the reverse order represented by chain-like network chunks "A", thus it can encode both past and future context of the sequence.

The Luong attention layer (Luong et al., 2015) is used to calculate the attention weights from the encoder outputs " $h_n$ " and the hidden state of the decoder from the current time step " $s_t$ ". These attention weights allow the decoder to focus on the most important parts in the input sentence and are calculated through "score functions" that use three different methods: dot, general and concat (see formula 4.2 that displays the equations for the different methods, where " $s_t$ " is the decoder hidden state from the current time step, " $h_n$ " are all encoder hidden states, while " $W_a$ " and " $v_a$ " are the model parameters for making the predictions based on the alignment vector). These functions are used to calculate the context vector "c" that is then used to make predictions for the next word by the decoder.

$$score(s_t, h_n) = \begin{cases} s_t^{\mathsf{T}} h_n & dot\\ s_t^{\mathsf{T}} W_a h_n & general\\ v_a^{\mathsf{T}} tanh(W_a[s_t; h_n]) & concat \end{cases}$$
(4.2)

The decoder GRU generates the output sentence one token at a time. It predicts the next token based on the context vector "c" and the current decoder hidden state  $s_n$ , until it outputs the "<EOS>" token, which represents the end of the sentence. The GRU used here is unidirectional (it has only one direction and handles the past context).



Fig. 4.5.: Architecture of the seq2seq encoder-decoder generative model with GRU RNNs and Self-Attention mechanism

To test the performance of different Luong score functions, five sentences from the user tests conducted during the first implementation (see Chapter 3) were selected: "Global warming will make winters a thing of the past", "Yes, but private investment will always want to improve negative features of gambling", "So therefore there exists no further alternative path to explore beyond educational programs?", "I argue that saving anyone goes against natural selection", "There are situations where you would have to choose which persons to save as all cant be saved, would it not be ok in this situation to let some die to save more people?". These user inputs were chosen because they could not match the intents in the first implementation and triggered the Fallback message. The second implementation of ArgueBot was set to prompt the generative model instead of the Fallback message. The results are presented in the table 4.6.

The general function was chosen for the generative model in the ArgueBot. Its generated sentences from the experiment were better composed than that of the other functions.

Sentence	dot	general	concat
Global warming	the global economy	a global warming is	global warming is
thing of the past	bad	priority	economic and devel-
		Priority	opment measure to
			have produced hu-
			man rights
Yes, but private in-	for example the	as the best way to	but what a lot of
vestment will al-	private sector has	progress is that it	money is going to
prove negative fea-	decades	whether she is be-	service based mar-
tures of gambling	uccuucb	ing spent or be-	ket
		cause of the risk	
		that is being spent	
Co thoustons thous	this is a nuchlam	on the condition	all this is trues
So therefore there exist no further al-	this is a problem	this is a clear ap-	will not offer that
ternative path to ex-	about what people	ilv be used malev-	all women are
plore beyond educa-	want to learn about	olently and is not	expected to abide
tional programs?	how they play a nat-	consistent	by
	ural process		
I argue that saving	i am sure it is a	i think that is ex-	in fact behes claim
natural selection	have to be scientifi-	correct theory	not only because
natural selection	cally scientifically	concer meory	they are unable to
			abuse their intel-
			ligence if i know
	1	.1	there is no god
There are situations	however if a person	the court s respon-	there is no reason
have to choose	tion those who are	sidility is to the	tain areas where in
which persons to	legally prepared to	late people from the	a year there would
save as all cant	use guns for crimi-	feeling of language	be no need for the
be saved, would it	nal purposes to be	and religion and is	vast majority of life
not be ok in this	able to save their	thus unacceptable	
situation to let	life and then put	in most cases when	
more people?		punished for society	
more people:		Pulliblica for society	

Tab. 4.6.: Comparison between text generations based on different Luong score functions

## 4.4 Conclusion

This chapter described the second implementation of the ArgueBot platform. The model of the chatbot was changed from retrieval-based to hybrid, combining both retrieval and generative models. This was done to compensate for the limitations of the retrieval model being restricted to the arguments stored in the database. The stance classification method was changed from using sentiment analysis as a tool for treating it as a binary classification problem and applying machine learning classification methods. After conducting a set of experiments, the LSTM with self-attention mechanism classifier was chosen to include in the chatbot.

Following research questions were answered within this chapter:

- SQ:1 *How can the model recognize and handle the arguments?* Alike the first implementation, Dialogflow was used to identify the context of the user input. If it was an argument, it was further handled by the retrieval model that retrieved the response from the database. If Dialogflow couldn't identify the meaning of the user input, it was handled by the generative model that created a seq2seg response.
- SQ:2 *How can stance classification be applied for the conversational agents*? In the second implementation polarity LSTM SAM was machine learning model was used to classify the stance.
- SQ:3 *What is an appropriate model for the chatbot's response generation?* The second implementation of the ArgueBot used the hybrid model that combined both retrieval and generative models for the response generation.

# Final evaluation of the ArgueBot

Users do not care about what is inside the box, as long as the box does what they need done.

> — **Jef Raskin** The Humane Interface (2001)

This chapter will describe the evaluation strategy for the second implementation of the ArgueBot which is also a final version. Similarly to the evaluation of the first implementation, the evaluation was conducted through the user tests via an updated platform (see figure 5.1) and a survey created in Google Forms (see Appendix B). In this chapter, first, an overview of the user tests will be given with the description of the process, the platform used and the survey questions. Then, the survey results will be presented with the help of graphs and statistical metrics (mean value with its standard deviation and t-test). As the stance classifier was changed in the middle of the user tests, the survey results were split accordingly and handled separately for the affected questions when conducting the analysis. This gave the opportunity to compare two different ArgueBot versions that used different stance classifiers (LSTM and LSTM SAM) and analyze if or how they affected the performance of the chatbot. Finally, conversation length will be presented with the help of turns and a graph.

### 5.1 Overview

Before conducting the user tests, the platform was tested by five colleagues in order to assure the test quality. These tests are not included in the results for the second implementation. Based on the feedback from these pilot tests, the decision was made to change the front page and include the instructions for the interaction with the chatbot. Some other minor changes were made for the main page as well, such as changing the function for the dropdown that included the list of the debates as it was not properly updating when changing the debate topic.

The user tests were conducted during one week in July. The link to the ArgueBot application was sent to the colleagues at Findwise via an internal channel. A

Facebook group with the link and information about the chatbot was created, where 355 acquaintances got invited. The user tests resulted in 38 survey answers. The low number of answers is most probably a result of bad timing and acquaintances being on vacation. One user test was removed from the resulting answers, as the user did not receive any response from the chatbot because of a server error and could therefore not evaluate the interaction with the chatbot fairly.

The ArgueBot platform included a front page with the information about the chatbot, instructions for the user test, and a consent form (see figure 5.1a). After agreeing to the terms of the user test, the users were redirected to the main page with the chatbot (see figure 5.1b). The main page was separated into two sections. On the left side, there was a menu for choosing a debate topic and browse between different debates available for that topic. Every debate had background information. The chatbot message interface was located on the right side of the page. The identification number, every user was assigned with, was located at the top of the page, while the link to the survey with a reminder to copy the identification code before proceeding was located at the bottom of the page.

The survey was created in Google Forms (see Apendix B) and included the same questions as the previous survey (for the first implementation see 3.3) with two differences:

- The users were asked if they had any previous interactions with any chatbot before with predefined yes/no answers: yes (ArgueBot and other chatbots), yes (ArgueBot), yes (other chatbots), no. This question provided some user background on the users' previous interactions with the chatbots.
- The users were asked to rate the response quality for the generated responses (marked with GR at the end of the sentence) and retrieved responses separately. Answering the question that asked the users to rate the response quality for the generated responses was optional, as not all the users received such responses.

As previously mentioned, the stance classifier was changed in the middle of the user tests after discovering that the previous classifier could not distinguish between different classes and predicted all the user inputs as the same class. The change of the classifier resulted in a split in the user test answers that were affected by the classifier. The user tests that used LSTM as a stance classier had 16 responses, while user tests with LSTM with self-attention mechanism (LSTM SAM) resulted in 21 responses (one survey response from 22 was removed as mentioned before).

#### Welcome to ArgueBot!

#### Information Brochure

This research is a part of master thesis, entitled "ArgueBot: Enabling a debate through a hybrid retrieval-generation-based chatbot". The aim of the research is to create a chatbot that is able to maintain a meaningful debate with users on various topics throughout the whole interaction. The goal of the chatbot is to inform, challenge and provoke the user into critical thinking.

The interaction with the chatbot will be saved for further analysis. Please, do not enter any private information into the chatbot such as your name or email.

- Your data collected by the chatbot:
- will not be passed on to third parties;
- will be treated confidentially;
  will only be used in anonymised form for this research;
  will be stored so that unauthorized persons cannot access it.

#### ArgueBot Instructions

In this research, you will interact with the chatbot in the debate on different topics. You are allowed to switch between different debates during the interaction.

Enter "Hi" or your argument to start the conversation.

The chatbot both retrieves sentences from the dataset and generates new sentences. You can differentiate the generated sentences by seeing "GR" at the end of the chatbot's response.

Be aware that the opinions stated by the chatbot do not represent its developer's opinions but are retrieved or generated from the dataset containing multiple arguments with different points of view. Some of these responses might be considered offensive.

You will be asked to fill in the survey about your interaction, please copy the identication code before proceeding.

By checking in this checkbox, you confirm your consent, that you hereby understood your rights and agree to continue. Continue

#### (a) The front page

First select a debate from here (it will take few seconds to load):	ArgueBot
digital freedoms \$	hi
not allow companies to collect/sell the personal data of their clients $\diamond$ The advent of computing has allowed governments, businesses and individuals to store, collate, analyse, and utilize huge amounts of data. Firms have made	Let's start the debate! I will argue that we collecting and selling personal information is a major violation of privacy. What is your opinion?
use of this power to store the personal data of their customers in order to better understand them and to cater to their needs. Others have even used the data online to offer targeted marketing aimed at the personal profiles of individuals, offen anarde human of information along drem ordering across in historiane at a	Today data is the power. Without data, there will be no future
that give them a window on the sorts of people using their services. Sometimes even phone numbers, email addresses, and physical addresses are sold in this way. Databrokers such as ChoicePoint and Acxiom purchase data from many	I disagree! Firms' data collection serves as a very real threat to individuals' privacy and identity on the internet.
sources, including government agencies, businesses, credit bureaus and investigators. They then offer "analytical services" on millions of households.	How about the service they provide based on your data?
This collation and utilization of data has spawned a secondary market in which firms have sold on their compiled data on individuals to other companies seeking to better understand the marketplace and to find consumers for their products. It is a booming \$300 billion-a-year industry.	I disagree! Anonymity is certainly not wholly guaranteed through these tactics, and the information is not entirely safe.
This vast increase in the dissemination and use of individuals' private data has resulted in extensive debates across the world, and has spawned many laws,	Enter your argument

#### (b) The main page

Fig. 5.1.: The interface of the ArgueBot for the second implementation

# 5.2 Survey results

This section presents the results of the survey and highlights the most important findings. The survey results were divided into two groups (two versions of the ArgueBot that used LSTM and LSTM-SAM as a stance classifier), but will jointly analyze them for survey responses that were not impacted by the stance classifier (such as grammar and debate background).

### 5.2.1 User Background

Figure 5.2 shows the distribution of users with different interaction backgrounds for the classifiers used. The distribution is quite similar, with both classifiers having users that had previous experience chatting with a chatbot. The LSTM SAM stance classifier ArgueBot version was tested by more users who previously tested ArguBot in the first implementation.



Fig. 5.2.: The distribution of different user backgrounds (previous interaction experience with chatbots) for different classifiers.

#### 5.2.2 Debate information

The purpose of the debate information presented on the left side of the platform was to give the users the context of the debate and help them to formulate their arguments. This area of the survey responses was handled together for both classifiers, as it stayed the same for both versions of ArgueBot.

All of the users (37 of 37) thought that the debate information was clearly presented and gave mostly positive feedback on its content. The users found it interesting and relevant. Moreover, 30 users of 37 found the debate information helpful. The users that did not find it helpful (7 of 37) commented that the information was too long or too heavy and suggested to either have a word cloud of the most important words or the outline with the major arguments. Some users who did not find it helpful preferred to directly dive into the debate with the chatbot instead of reading the background information first. This confirms the findings from the user tests for the first implementation of the ArgueBot that showed positive feedback for the background information's presence on the platform.

#### 5.2.3 Grammar

The users were asked to rate the grammar used by the chatbot on a scale from 1 to 10 (where 1 is very bad and 10 is very good). The survey answers for this question as the previous question were handled together for both classifiers. The reason for that is that the grammar of the responses was not affected by the stance classifier. Figure 5.3 shows how the users rated the grammar of the chatbot's responses (both retrieved and generated). The majority of the users rated the grammar higher than 7 (22 users from 37) with an average of  $7.49 \pm 1.8$ . Users gave mostly good feedback for the grammar used by the chatbot, apart from the generated sentences that 8 from 37 users indicated had poor grammar in the open-ended question. 27 of 37 users (73%) of the participants got the generated responses. In retrospect, this question could also have been separated for the generated and retrieved responses, which could help analyze the performance of the models with regard to the grammar of their responses.



Fig. 5.3.: How the users rated the grammar of the chatbot's responses

#### 5.2.4 Conversation flow

The users were asked to rate how natural (human-like) the conversation flow with the chatbot felt on the scale from 1 to 10 (1 is unnatural, 10 is natural). This question was handled separately for different classifiers used.

Figure 5.4 shows user ratings for the naturalness of the conversation flow with the chatbot. The blue color shows the results for the ArgueBot version that used the

LSTM stance classifier, and the orange color represents the results for the chatbot version that used the LSTM SAM as a stance classifier. Ratings for both classifiers had a distribution towards the lower ratings, with the majority of the votes for rating 3 and the average of  $4.56 \pm 2.03$  for the LTSM stance classifier and  $3.38 \pm 1.94$  for the LSTM SAM stance classifier.

The t-test is a statistical hypothesis test that was used to determine if there was any significant difference between the means of two different classifiers for the conversation flow's naturalness. The significance level was set to 0.05 (the most commonly used significance level that was used to compare the t-test value with). According to the t-test, that showed 0.08, there is a slightly significant difference between the results for the naturalness of the conversation flow for the different classifiers used.

Users felt that the chatbot did not understand their inputs. For the LSTM classifier, the users said that it was hard to understand the chatbot's stance, as it always replied with either "I agree" or "I disagree" to every user argument. For the LSTM SAM stance classifier, the users felt that the chatbot's responses were unrelated to what the users said and having "I agree"/"I disagree" statements felt rehearsed and broke the continuity of the conversation. One user suggested to add statements of the type "I agree but... " to improve the conversation flow.



**Fig. 5.4.:** User ratings for how natural (human-like) the conversation flow with the chatbot felt, where 1 is unnatural and 10 is natural. The percentage score shows the distribution for the rating amongst the users for the ArgueBot with LSTM (blue color) and LSTM SAM (orange color) as a stance classifier

### 5.2.5 Response quality

The rating of the response quality for the chatbot was separated into two questions: one for generated sentences created by the Generative Model (see Chapter 4.3) and one for the retrieved sentences that used the Retrieval Model to compare the user input with the arguments stored in the database (see Chapter 3.2.2). These

questions were handled separately for the different classifiers used, and can be seen in Figure 5.5 for the generated sentences and Figure 5.6 for the retrieved sentences. Even though the generated sentences were not affected by the change of the classifier, their ratings could still indirectly be affected by the performance of the Retrieval Model that used the classifier. The users could, for example, rate the retrieved sentences higher not because their quality was better, but because the quality of the generated responses was worse. The ArgueBot that used the LSTM stance classifier is represented in blue, while the ArgueBot version that used the LSTM SAM is represented in orange.

The quality of the generated sentences was rated as unsatisfactory by the majority of the respondents for both classifiers with an average of  $3.17 \pm 2.04$  for the LSTM version and  $3.13 \pm 2.85$  for the LSTM SAM version. The t-test showed 0.97, there is, therefore, a statistically significant difference between the classifiers for the rating of the generated responses.

The users thought that the generated sentences made little to no sense. Some of the users stated that the generated responses were very random and did not relate to the topic discussed.



**Fig. 5.5.:** User ratings for the response quality of the generated sentences, where 1 is unsatisfactory and 10 is satisfactory. The percentage score shows the distribution for the rating amongst the users answering the question for the ArgueBot with LSTM (blue color) and LSTM SAM (orange color) as a stance classifier.

The quality of the retrieved sentences was rated on average  $5.14 \pm 2.35$  for the LSTM SAM version and  $4.81 \pm 2.14$  for the LSTM version. The t-test for the retrieved responses showed 0.66, there is, therefore, a statistically significant difference between the results for the different classifiers used compared to the 0.05 significance level.

The respondents that tested the LSTM version of the ArgueBot felt that the chatbot didn't understand them, that there was no common thread throughout the responses and that the responses were often unrelated to what the user said. The respondents



that tested the LSTM SAM version felt that the retrieved responses were better than generated sentences and more related to the debate but still limited.

**Fig. 5.6.**: User ratings for the response quality of the retrieved sentences, where 1 is unsatisfactory and 10 is satisfactory. The percentage score shows the distribution for the rating amongst the users for the ArgueBot with LSTM (blue color) and LSTM SAM (orange color) as a stance classifier.

The difference between the user rating for the retrieved and the generative responses can be seen in figure 5.7 for both classifiers together. The figure shows that retrieved responses marked with violet color (the average is  $5 \pm 2.24$ ) were rated relatively better than the generated responses marked with pink color (the average is  $3.15 \pm 2.48$ ). The t-test showed 0.003, which compared to the significance level 0.05 showed no statistically significant difference between the responses.



**Fig. 5.7.:** User ratings for the response quality of the generated responses (pink color) versus retrieved responses (violet color), where 1 is unsatisfactory and 10 is satisfactory. The percentage score shows the distribution for the rating amongst the users answering the question for the ArgueBot for both classifiers

# 5.3 Conversation length

When it comes to the conversation length, a number of turns was used as a metric. This metric shows how many input/output pairs the conversations have (one user could have had multiple conversations on different debate topics). The conversation

length can be an indicator of how engaging the conversation with the chatbot is perceived by the user (the longer it is, the more engaging it is perceived).

Figure 5.8 shows the number of turns per conversation for the two different stance classifiers used in the second implementation of the ArgueBot. One conversation here is delimited to one specific debate. An average number of turns for the LSTM classifier is  $8 \pm 6$  and for the LSTM SAM  $7 \pm 8$ . The standard deviation for both classifiers is higher than their means, which means the high spread of the data. The decision was, therefore, made to group the number of turns in the range of five to understand the data better.

The t-test could not be performed here, as the number of turns were previously grouped. It was therefore not possible to determine if there was any statistically significant difference between different classifiers. But according to the figure 5.8, an assumption can be made that there is a similar distribution of the grouped number of turns for both classifiers with the majority of conversations being less than 10 turns.



**Fig. 5.8.**: Comparison between different stance classifiers for number of turns per conversation, where blue represents LSTM stance classifier and orange LSTM SAM stance classifier. The percentage score shows the distribution of the range for a number of turns for all the conversations for that classifier

# 5.4 Conclusion

As the mean value from the survey results showed, the response quality of the retrieved sentences was only slightly higher rated for the ArgueBot version using the LSTM with self-attention mechanism stance classifier. There is a statistically significant difference between the two classifiers in terms of the naturalness of the conversation flow, generated response quality and retrieved response quality. But there is no statistically significant difference between the generated and the retrieved responses counted for the survey results for both classifiers together.

Following research questions was answered within this chapter:

SQ:5 How can such a chatbot be evaluated? The ArgueBot platform was updated to perform the user tests in the combination with the survey that collected user's feedback on their interaction with the chatbot and its performance. The survey results were analyzed with the help of graphs, mean value with its standard deviation value and t-test. Moreover, the users' engagement with the chatbot was evaluated with the help of the number of turns.

# Discussion

If we are willing to make such efforts in order to understand foreign cultures, unknown species and distant planets, it might be worth working just as hard in order to understand our own minds. And we had better understand our minds before the algorithms make our minds up for us.

> — Yuval Noah Harari 21 Lessons for the 21st Century (2018)

This chapter will discuss the results of the study, its limitations, and factors that could have influenced the results. The limitations are divided into subsections for the ArgueBot, the stance classification, the generative model, and the hybrid model.

### 6.1 ArgueBot

The dataset used by the ArgueBot was the ArguAna Counterargs corpus (Wachsmuth et al., 2018) that consists of high-quality arguments with different points of view for a wide range of the debates. These arguments are not adjusted to conversational interfaces such as chatbots and therefore presented several challenges.

Firstly, the debate authors used multiple types of citations and references. In order to clean the data and only retrieve arguments, as well as prepare the datasets for machine learning models, multiple regular expressions were used to match different referencing styles used in the arguments. Even after extensive cleaning, the datasets had to be manually reviewed to remove the rest of the references that the regular expressions were not able to capture. Another challenge was to change the names of the debates from "This House would..." wording format into more user-friendly names. This dataset was not used before in a chatbot (to the best of my knowledge) setting and was both challenging and interesting to work with.

The retrieval model used by the chatbot used cosine similarity to compare how similar the user input is to the main claim of every argument for the chosen debate.

The main claim of the argument was one sentence consisting of 3-10 words (see an example of a debate structure in Figure 3.1 with "PRO/CON + number" main claims). This was done with the assumption that the user input would also be 3-10 words long and the similarity between these can, therefore, be extracted. The model then chose the argument of which the main claim had the highest similarity with the user utterance. Sometimes this worked as intended, but sometimes the user utterance differed from all the main claims available for that debate, or the main claim available in the dataset was not descriptive enough. The latter case resulted in the model retrieving the argument with the highest similarity but with no actual relevance to the user input. A better approach might be to extract the most used words (debate-specific words) in every argument (separately for a point and a counterpoint) and use them as keywords in combination with the main claim to compare with the user input.

Another problem occurred when two different user utterances were matched to the same argument. The model then assumed that it should continue to retrieve sentences from the same argument and removed the used sentence from the database. In the first implementation when the user input was matched to an argument that didn't have any sentences left to retrieve, the chatbot would reply with the default message: "You already used this argument", which according to the user tests felt unnatural for many users as they used new arguments and the chatbot reply was false. For the second implementation, the retrieval strategy was slightly changed, with the model handling two arguments with the highest similarity to the user input, and if the first argument was empty, retrieving the next sentence from the second argument instead. This strategy, while fixing the issue with keeping replying with new sentences, did not fix the issue of the responses being irrelevant to the user input. To improve the relevance of the chatbot's retrieved responses, two strategies can be applied: extend the dataset with more arguments and improve the similarity computations. Cosine similarity was chosen because of its compatibility with the spaCy library, but other methods could have shown better performance, such as Word Mover Distance, universal sentence encoder, and Siamese Manhattan LSTM model (Adrien Sieg, 2019).

Correct context understanding (the meaning of the user input) is another challenging problem. To help the model understand the context of the user input, the Dialogflow service was used. The model extracted the 300 most used words in the debate and fed them in their lemma word form into Dialogflow as entities, that were later used by Dialogflow to classify the user input as an argument. This method does not take synonyms into consideration, and the context detection was therefore limited to the words used in the dataset for that debate. One solution could be to combine the most used words in the debate with their synonyms. Moreover, the ArgueBot hybrid model was built with the assumption that the user would input only arguments. It could therefore not handle complex questions that many users asked for example to clarify some term or statement used by the chatbot. It could not distinguish between statements and questions, which caused wrong agent responses. One solution could be to apply a self-attention mechanism on the user input in combination with the previous agent responses. Zhou et al. (2018) proposed DAM networks that used attention mechanism to capture sentence-level dependencies in a multi-turn chatbot. This solution can, therefore, be applied to improve the user engagement with the chatbot, as it is capable of taking into consideration the user's previous inputs.

### 6.2 Stance Classification

In the second implementation, the ArgueBot used machine learning techniques for stance classification. The stance classifier was trained on a dataset composed of sentences with their corresponding stances from the ArguAna Counterargs corpus (Wachsmuth et al., 2018) and the IBM Debater Claim Stance Dataset (Bar-Haim et al., 2017). The machine learning methods used were the LSTM and the LSTM with self-attentions, that focused on the most important words in the sequence. The classifier was used to predict the stance of the user input, which was then compared to the chatbot's stance that later replied with either "I agree" or "I disagree" statement. This agreement statement was included at the beginning of the ArgueBot's retrieved response. The classifier is therefore limited to the available arguments for training and is not topic dependent. If more time for the project would be available, this problem could be explored more. For example, it would be interesting to combine a machine learning classifier with feature extraction strategies suggested by Mandya et al. (2016) and sentiment towards the main claim as was suggested by Bar-Haim et al. (2017).

The user tests showed that the two stance classifiers had a statistically significant difference in the users' interaction with the chatbot in relation to conversation flow naturalness, and response quality for generated and retrieved sentences. Conversation flow and quality of generative responses were rated slightly higher for the LSTM version than LSTM SAM version of the ArgueBot. This can depend on several factors, for example, different rating understanding for different users. At the same time, the generated responses were depended on the performance of the classifier only in relation to the quality of retrieved responses. It can, therefore, have an opposite effect: the LSTM version was rated higher for the quality of generated responses that used LSTM classifier performed worse. This agrees with the conclusion that the quality of the LSTM SAM version for the

retrieved responses had a statistically significant difference and was rated higher than the LSTM version. It can, therefore, be concluded, that LSTM SAM classifier performed better and improved the quality of the retrieved responses. At the same time, it did not improve the naturalness of the conversation flow. A conclusion can, therefore, be made that the stance classification cannot be seen as a simple binary classification problem. The stance classification was shown to be a more complex problem that needs a more complex model for the classification than, unfortunately, was out of the scope for this project.

The overall lesson learned from the user tests is that the stance classifier should be more reliable in order to improve the user experience. If the stance classifier predicts wrong and results in the wrong agreement statement, the user assumes that the chatbot did not understand their argument, which reduces their satisfaction of the interaction with the chatbot.

### 6.3 Generative Model

The generation of sentences is also a very interesting and challenging problem. When done properly, it can enrich the conversation with the chatbot with always new arguments to debate on. When done poorly, it can damage the conversation flow and confuse the user instead. Unfortunately, the latter was the case for the ArgueBot. According to the user tests, generated sentences were often unrelated to the debate topic, sometimes grammatically incorrect and did not make sense in the debate context.

Because of Google Colab's limitations, that was used to pre-train all the machine learning models, the dataset used was not enough to generate sentences that made more sense. Google Colab and a smaller dataset were used because it was fast and convenient. If more time and computational power for the project were available, it would be interesting to train a model on the larger dataset composed of for example the Internet Argument Corpus (IAC)<sup>1</sup> and compare the quality of the generated sentences.

# 6.4 Hybrid Model

It is important to keep in mind that although the generative models can build understandable and grammatically correct responses given the conversation context, they are likely to return general responses. One suggestion to overcome this issue can

<sup>&</sup>lt;sup>1</sup>https://nlds.soe.ucsc.edu/iac2

be to use personalized generated responses recently developed by Microsoft research team through a conversation modeling toolkit Icecaps (Shiv et al., 2019). Their toolkit helps to build agents with induced personalities that are able to generate diverse responses for different users.

While retrieval-based models are prompted to give more relevant responses than the generative-based models, they still lack the flexibility of the latter and are limited to the stored information. The proposed hybrid model in this research included both models for the ArgueBot response generation but did not succeed in overcoming these limitations. This might be the reason why the majority of the users rated the chatbot's conversation flow as unnatural and why there was no statistically significant difference between the quality of retrieved and generated responses. The project shows great potential, but the ArgueBot still has a long way to go before becoming a reliable debate partner.

# Conclusion

Our future is a race between the growing power of our technology and the wisdom with which we use it. Let's make sure that wisdom wins.

> — **Stephen Hawking** Brief Answers to the Big Questions (2018)

This study aimed to create a platform where the users can maintain a meaningful debate on various topics with a chatbot. Even though the chatbot had troubles with context understanding, correct stance classification and generating new meaningful arguments, many users still found the interaction fun, interesting, and meaningful.

In order to answer the problem statement of the study, namely "*How can a hybrid retrieval-generation-based chatbot maintain a debate with a user for various topics*?", the sub-questions were answered:

- SQ:1 *How can the model recognize and handle the arguments?* The Arguebot used the Dialogflow's intent and entities to analyze the user input and determine whether it is an argument or not. Dialogflow's intent detection functionalities were used to understand the context of the user input and select the appropriate response strategy. It could detect arguments, questions for clarification, the agent's stance and small talk. When the intent could not be matched, the generative model was activated. If the intent was matched with the argument intent, the user input was further analyzed by the stance classifier and the retrieval model. This was proven to work well when the user used the same words as were stored in the entities but failed when the user deviated from the intended conversation flow (for example when they asked more specific questions, the meaning of some terms and so on).
- SQ:2 *How can stance classification be applied for the conversational agents?* Both classification models used in the Arguebot, a model with sentiment analysis and the binary machine learning model, were not performing well enough for the conversational agent. A more advanced model for stance classification is, therefore, needed.

- SQ:3 *What is the appropriate model for the chatbot's response generation?* The Argue-Bot used a hybrid model that combined both generative and retrieval models. The generative model used a self-attention mechanism to generate new arguments when the retrieval model could not be applied. The generated sentences often did not have good quality and were unrelated to the specific debate topic. When given more time and computational power, the generative model can be pre-trained with more data during more time, which might improve its performance. This was unfortunately out of scope for this study.
- SQ:4 *How can human-like conversation with the chatbot be carried out in the debate domain?* The ArgueBot used Dialogflow's pre-trained agent for small talk. Stance classification was used to react on the user input with either agreement or disagreement. In retrospect, the stance classifier performed not well enough and decreased the naturalness of the conversation flow. The debate is often more complex than just agreeing and disagreeing with the opponent, for example including the arguments of the "I agree, but..." structure.
- SQ:5 *How can such a chatbot be evaluated?* The evaluation of the ArgueBot was conducted through the user tests via a created platform and a survey created in Google Forms. The survey results were analyzed with the help of graphs and statistical metrics (mean value with its standard deviation and t-test). Moreover, the user engagement was evaluated in relation to the number of turns within each conversation that users had.

The hybrid retrieval-generation-based chatbot can maintain a debate with a user for various topics by creating an engaging, human-like experience. The agent should recognize the context of the user input, correctly classify the user's stance and provide with the relevant responses, whether they are retrieved or generated.

The next step for the ArgueBot would be to improve the conversation flow through a better similarity algorithm, context understanding, extended dataset and better classifier and generative model. To improve the similarity, keywords extracted from debate-specific words can in combination with the main claim be used for comparison with the user input. The knowledge base of the ArgueBot can be extended with more conversational data. The stance classifier can be improved by combining a machine learning classifier with feature extraction strategies and sentiment towards the main claim. The generative model can be improved by training it during a longer period on a bigger amount of data. Moreover, other models can be used to make the interaction with the chatbot more engaging, such as DAM attention mechanism and Microsoft's Icecaps. I believe given these improvements, the Arguebot can become an interesting debate partner, broaden the political discussion, and promote critical thinking in many.

# Bibliography

- Abbott, Rob, Brian Ecker, Pranav Anand, and Marilyn A Walker (2016). "Internet Argument Corpus 2.0: An SQL schema for Dialogic Social Media and the Corpora to go with it." In: *LREC* (cit. on p. 8).
- Bar-Haim, Roy, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim (2017). "Stance classification of context-dependent claims". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume* 1, Long Papers, pp. 251–261 (cit. on pp. 30, 56).
- Cahn, Jack (2017). "CHATBOT: Architecture, design, & development". In: University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science (cit. on p. 7).
- Carstens, Lucas and Francesca Toni (2015). "Towards relation based argumentation mining". In: *Proceedings of the 2nd Workshop on Argumentation Mining*, pp. 29–34 (cit. on p. 39).
- Carstens, Lucas and Francesca Toni (2017). "Using argumentation to improve classification in natural language problems". In: *ACM Transactions on Internet Technology (TOIT)* 17.3, p. 30 (cit. on p. 39).
- Chapman, Graham and Monty Python (1989). *The Complete Monty Python's Flying Circus: All the Words. Volume one.* Vol. 1. Pantheon, p. 86 (cit. on p. 4).
- Cho, Kyunghyun, Bart van Merriënboer, Caglar Gulcehre, et al. (2014). "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*). Doha, Qatar: Association for Computational Linguistics, pp. 1724–1734 (cit. on p. 40).
- Habernal, Ivan and Iryna Gurevych (2017). "Argumentation mining in user-generated web discourse". In: *Computational Linguistics* 43.1, pp. 125–179 (cit. on p. 5).
- Higashinaka, Ryuichiro, Masahiro Mizukami, Hidetoshi Kawabata, et al. (2018). "Role play-based question-answering by real users for building chatbots with consistent personalities".
  In: *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 264–272 (cit. on p. 10).
- Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780 (cit. on p. 37).
- Huang, Anna (2008). "Similarity measures for text document clustering". In: *Proceedings* of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand. Vol. 4, pp. 9–56 (cit. on p. 15).

- Keller, Thomas E., James K. Whittaker, and Tracey K. Burke (2001). "Student Debates in Policy Courses: Promoting Policy Practice Skills and Knowledge through Active Learning". In: pp. 343–355 (cit. on p. 1).
- Kim, Jintae, Hyeon-Gu Lee, Harksoo Kim, Yeonsoo Lee, and Young-Gil Kim (2018). "Two-Step Training and Mixed Encoding-Decoding for Implementing a Generative Chatbot with a Small Dialogue Corpus". In: Proceedings of the Workshop on Intelligent Interactive Systems and Language Generation (2IS&NLG), pp. 31–35 (cit. on p. 6).
- Klopfenstein, Lorenz Cuno, Saverio Delpriori, Silvia Malatini, and Alessandro Bogliolo (2017). "The rise of bots: A survey of conversational interfaces, patterns, and paradigms". In: *Proceedings of the 2017 Conference on Designing Interactive Systems*. ACM, pp. 555–565 (cit. on p. 2).
- Kuhn, Deanna (1991). The Skills of Argument. Cambridge University Press (cit. on p. 4).
- Lawrence, John and Chris Reed (2014). "AIFdb Corpora." In: *COMMA*, pp. 465–466 (cit. on p. 39).
- Le, Dieu-Thu, Cam Tu Nguyen, and Kim Anh Nguyen (2018). "Dave the debater: a retrievalbased and generative argumentative dialogue agent". In: *Proceedings of the 5th Workshop on Argument Mining*, pp. 121–130 (cit. on pp. 6–9).
- Levy, Ran, Shai Gretz, Benjamin Sznajder, et al. (2017). "Unsupervised corpus–wide claim detection". In: *Proceedings of the 4th Workshop on Argument Mining*, pp. 79–84 (cit. on pp. 5, 6).
- Lin, Zhouhan, Minwei Feng, Cicero Nogueira dos Santos, et al. (2017). "A structured selfattentive sentence embedding". In: *CoRR* abs/1703.03130. arXiv: 1703.03130 (cit. on p. 37).
- Lippi, Marco and Paolo Torroni (2016). "Argumentation mining: State of the art and emerging trends". In: *ACM Transactions on Internet Technology (TOIT)* 16.2, p. 10 (cit. on p. 4).
- Luong, Thang, Hieu Pham, and Christopher D. Manning (2015). "Effective Approaches to Attention-based Neural Machine Translation". In: *Proceedings of the 2015 Conference* on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, pp. 1412–1421 (cit. on p. 40).
- Ma, Wenjia, WenHan Chao, Zhunchen Luo, and Xin Jiang (2018). "CRST: a claim retrieval system in Twitter". In: *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pp. 43–47 (cit. on p. 6).
- Mandya, Angrosh, Advaith Siddharthan, and Adam Wyner (2016). "Scrutable feature sets for stance classification". In: *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pp. 60–69 (cit. on pp. 6, 56).
- Manning, Christopher D (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press (cit. on pp. 6, 8).
- Moore, Robert J, Raphael Arar, Guang-Jie Ren, and Margaret H Szymanski (2017). "Conversational UX design". In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, pp. 492–497 (cit. on p. 2).
- Müller, Andreas C, Sarah Guido, et al. (2016). *Introduction to machine learning with Python: a guide for data scientists.* " O'Reilly Media, Inc." (cit. on pp. 32, 33).

- Rakshit, Geetanjali, Kevin K Bowden, Lena Reed, Amita Misra, and Marilyn Walker (2019)."Debbie, the debate bot of the future". In: *Advanced Social Interaction with Agents*. Springer, pp. 45–52 (cit. on pp. 7, 8).
- Sedoc, Joao, Daphne Ippolito, Arun Kirubarajan, et al. (2018). "ChatEval: A Tool for the Systematic Evaluation of Chatbots". In: Proceedings of the Workshop on Intelligent Interactive Systems and Language Generation (2IS&NLG), pp. 42–44 (cit. on p. 9).
- Shiv, Vighnesh Leonardo, Chris Quirk, Anshuman Suri, et al. (2019). "Microsoft Icecaps: An Open-Source Toolkit for Conversation Modeling". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 123– 128 (cit. on p. 58).
- Shum, Heung-Yeung, Xiao-dong He, and Di Li (2018). "From Eliza to XiaoIce: challenges and opportunities with social chatbots". In: *Frontiers of Information Technology & Electronic Engineering* 19.1, pp. 10–26 (cit. on p. 9).
- Slonim, Noam and Ranit Aharonov (2018). "Proceedings of the 5th Workshop on Argument Mining". In: *Proceedings of the 5th Workshop on Argument Mining* (cit. on p. 4).
- Sokolova, Marina, Nathalie Japkowicz, and Stan Szpakowicz (2006). "Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation". In: *Australasian joint conference on artificial intelligence*. Springer, pp. 1015–1021 (cit. on p. 33).
- Somasundaran, Swapna and Janyce Wiebe (2010). "Recognizing stances in ideological on-line debates". In: *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. Association for Computational Linguistics, pp. 116–124 (cit. on p. 4).
- Stab, Christian, Johannes Daxenberger, Chris Stahlhut, et al. (2018). "ArgumenText: Searching for Arguments in Heterogeneous Sources". In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pp. 21–25 (cit. on p. 6).
- Swanson, Reid, Brian Ecker, and Marilyn Walker (2015). "Argument Mining: Extracting Arguments from Online Dialogue". In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue. Prague, Czech Republic: Association for Computational Linguistics, pp. 217–226 (cit. on p. 8).
- Tammewar, Aniruddha, Monik Pamecha, Chirag Jain, Apurva Nagvenkar, and Krupal Modi (2018). "Production Ready Chatbots: Generate if not Retrieve". In: *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence* (cit. on p. 7).
- Turing, A. M. (1950). "I.—COMPUTING MACHINERY AND INTELLIGENCE". In: Mind LIX.236, pp. 433-460. eprint: http://oup.prod.sis.lan/mind/article-pdf/LIX/ 236/433/9866119/433.pdf (cit. on p. 9).
- Wachsmuth, Henning, Martin Potthast, Khalid Al Khatib, et al. (2017). "Building an argument search engine for the web". In: *Proceedings of the 4th Workshop on Argument Mining*, pp. 49–59 (cit. on p. 6).
- Wachsmuth, Henning, Shahbaz Syed, and Benno Stein (2018). "Retrieving the Best Counterargument without Prior Topic Knowledge". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Melbourne, Australia: Association for Computational Linguistics (cit. on pp. 6, 11, 12, 25, 30, 38, 39, 54, 56).

- Walker, Marilyn A, Jean E Fox Tree, Pranav Anand, Rob Abbott, and Joseph King (2012). "A Corpus for Research on Deliberation and Debate." In: *LREC*. Istanbul, pp. 812–817 (cit. on p. 39).
- Walton, Douglas (2009). "Argumentation theory: A very short introduction". In: *Argumentation in artificial intelligence*. Springer, pp. 1–22 (cit. on p. 4).
- Weizenbaum, Joseph et al. (1966). "ELIZA—a computer program for the study of natural language communication between man and machine". In: *Communications of the ACM* 9.1, pp. 36–45 (cit. on p. 9).
- Yang, Liu, Junjie Hu, Minghui Qiu, et al. (2019). "A hybrid retrieval-generation neural conversation model". In: *arXiv preprint arXiv:1904.09068* (cit. on p. 7).
- Zhou, Xiangyang, Lu Li, Daxiang Dong, et al. (2018). "Multi-turn response selection for chatbots with deep attention matching network". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 1118–1127 (cit. on p. 56).
- Zhu, Pengfei, Zhuosheng Zhang, Jiangtong Li, Yafang Huang, and Hai Zhao (2018). "Lingke: a Fine-grained Multi-turn Chatbot for Customer Service". In: *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*. Santa Fe, New Mexico: Association for Computational Linguistics, pp. 108–112 (cit. on pp. 6, 7).

# Footnotes

- Adrien Sieg (2019). Text Similarities : Estimate the degree of similarity between two texts. URL: https://medium.com/@adriensieg/text-similarities-da019229c894 (visited on June 28, 2019) (cit. on p. 55).
- ChatEval (2019). URL: https://chateval.org/ (visited on Mar. 4, 2019) (cit. on p. 9).
- Dialogflow (2019). URL: https://dialogflow.com/ (visited on Mar. 4, 2019) (cit. on pp. 9, 17).
- Doha Debates (2019). Why This Debate and Why Now? URL: https://dohadebates.com/ (visited on Apr. 2, 2019) (cit. on p. 1).
- Findwise (2019). Search Driven Solutions. URL: https://findwise.com/en (visited on Apr. 2, 2019) (cit. on p. 2).
- Github (2019). Dialogflow: Python Client, Github. URL: https://github.com/googleapis/ dialogflow-python-client-v2 (visited on June 10, 2019) (cit. on p. 17).
- IBM Research AI (2019). Project Debater. URL: https://www.research.ibm.com/ artificial-intelligence/project-debater/ (visited on Feb. 6, 2019) (cit. on p. 1).
- idebate (2019). International Debate Education Association. URL: https://idebate.org/ (visited on Apr. 4, 2019) (cit. on p. 11).
- kaggle (2019). Quora Question Pairs. URL: https://www.kaggle.com/c/quora-questionpairs (visited on Apr. 8, 2019) (cit. on p. 8).
- Matthew Inkawhich (2019). *Chatbot tutorial*. URL: https://pytorch.org/tutorials/ beginner/chatbot\_tutorial.html (visited on June 10, 2019) (cit. on p. 39).
- Merriam-Webster (2019). Debate. URL: https://www.merriam-webster.com/thesaurus/ debate (visited on Feb. 12, 2019) (cit. on p. 1).
- Ngrok (2019). URL: https://ngrok.com/ (visited on July 18, 2019) (cit. on p. 18).
- NLTK (2019). Natural Language Toolkit. URL: https://www.nltk.org/ (visited on Mar. 21, 2019) (cit. on p. 15).
- Numpy (2019). Numpy Homepage. URL: https://www.numpy.org/ (visited on July 17, 2019) (cit. on p. 16).
- Pallets (2019). Flask SQLAlchemy. URL: https://flask-sqlalchemy.palletsprojects. com/en/2.x/ (visited on Apr. 8, 2019) (cit. on p. 18).

- Pandorabots (2019). Pandorabots Documentation. URL: https://pandorabots.com/docs/ (visited on Mar. 4, 2019) (cit. on p. 9).
- Panetta, K. (2019). Gartner Top 10 Strategic Technology Trends for 2019. URL: https://
  www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technologytrends-for-2019/ (visited on Feb. 6, 2019) (cit. on p. 2).
- Prakash Pandey (2019). Text Classification Pytorch. URL: https://github.com/prakashpandey9/ Text-Classification-Pytorch (visited on June 8, 2019) (cit. on p. 31).
- Projects, The Pallets (2019). *Flask*. URL: https://palletsprojects.com/p/flask/ (cit. on p. 18).
- PyTorch (2019). URL: https://pytorch.org/ (visited on July 26, 2019) (cit. on p. 31).
- spaCy (2019). Industrial-Strength Natural Language Processing. URL: https://spacy.io/ (visited on Mar. 21, 2019) (cit. on pp. 15, 30, 38).
- SQLite (2019). SQLite Homepage. URL: https://www.sqlite.org/index.html (visited on July 17, 2019) (cit. on p. 15).
- UC Santa Cruz (2019). Natural Language and Dialogue Systems: Internet Argument Corpus. URL: https://nlds.soe.ucsc.edu/iac2 (visited on Mar. 21, 2019) (cit. on pp. 39, 57).
- Wikipedia (2019). *Ethics of artificial intelligence*. URL: https://en.wikipedia.org/wiki/ Ethics\_of\_artificial\_intelligence (visited on Mar. 20, 2019) (cit. on p. 5).

# List of Figures

3.1	An example of a debate's architecture	13
3.2	Architecture of first implementation	14
3.3	A conversation conducted during the user tests for the first implementa- tion of the ArgueBot on the debate topic "Raise the school leaving age	
	to 18"	19
3.4	The interface of the ArgueBot for the first implementation	22
4.1	Architecture of second implementation	27
4.2	A conversation conducted during the user tests for the second implemen- tation of the ArgueBot on the debate topic "Making voting compulsory"	28
4.3	Comparison between two dialogues using different classifiers for the same debate topic "Banning the development of genetically modified organisms", where the blue color represents the user and the green	20
	color represents the agent	36
4.4	Architecture of the LSTM with Self-Attention Mechanism	38
4.5	Architecture of the seq2seq encoder-decoder generative model with	
	GRU RNNs and Self-Attention mechanism	41
5.1	The interface of the ArgueBot for the second implementation	46
5.2	The distribution of different user backgrounds (previous interaction	
	experience with chatbots) for different classifiers	47
5.3	How the users rated the grammar of the chatbot's responses	48
5.4	User ratings for how natural (human-like) the conversation flow with	
	the chatbot felt, where 1 is unnatural and 10 is natural. The percentage	
	score shows the distribution for the rating amongst the users for the	
	ArgueBot with LSTM (blue color) and LSTM SAM (orange color) as a	
	stance classifier	49
5.5	User ratings for the response quality of the generated sentences, where	
	1 is unsatisfactory and 10 is satisfactory. The percentage score shows	
	the distribution for the rating amongst the users answering the question	
	for the ArgueBot with LSTM (blue color) and LSTM SAM (orange color)	
	as a stance classifier.	50
5.6	User ratings for the response quality of the retrieved sentences, where	
-----	---	----
	1 is unsatisfactory and 10 is satisfactory. The percentage score shows	
	the distribution for the rating amongst the users for the ArgueBot with	
	LSTM (blue color) and LSTM SAM (orange color) as a stance classifier.	51
5.7	User ratings for the response quality of the generated responses (pink	
	color) versus retrieved responses (violet color), where 1 is unsatisfactory	
	and 10 is satisfactory. The percentage score shows the distribution for	
	the rating amongst the users answering the question for the ArgueBot	
	for both classifiers	51
5.8	Comparison between different stance classifiers for number of turns	
	per conversation, where blue represents LSTM stance classifier and	
	orange LSTM SAM stance classifier. The percentage score shows the	
	distribution of the range for a number of turns for all the conversations	
	for that classifier	52

### List of Tables

3.1	Distribution of debates, points, and counters over the topics in the	
	dataset (Wachsmuth et al., 2018)	12
3.2	System's response model with some examples from the dialogue on the	
	debate topic "Make voting compulsory"	20
4.1	Distribution of debates, points, and counters over the topics in the	
	database for the second implementation	26
4.2	System's response model with some examples from the dialogue on the	
	debate topic "Make voting compulsory"	29
4.3	The Dataset used for the Stance Classification in number of lines	30
4.4	Comparison between different machine learning models for stance	
	classification	33
4.5	Comparison between different machine learning models for stance	
	classification in the second experiment, where "c" stands for "con", "p"	
	for "pro" and "n" for "none"	34
4.6	Comparison between text generations based on different Luong score	
	functions	42

## Α

### Appendix Survey ArgueBot 1.0

#### ArgueBot

This research is a part of master thesis, entitled "ArgueBot: Enabling a debate through a multi-turn retrieval-based chatbot". The aim of the research is to create a chatbot that is able to maintain a meaningful debate with users on various topics throughout the whole interaction. The goal of the chatbot is to inform, challenge and provoke the user into critical thinking.

In this research, you will interact with the chatbot in the debate on different topics. Be aware that the opinions stated by the chatbot do not represent its developer's opinions in any way but are retrieved from the dataset containing multiple arguments with different points of view.

The interaction with the chatbot will be saved for further analysis. Please, do not enter any private information into the chatbot such as your password or email.

The data collected by the chatbot and this survey:

- will not be passed on to third parties;
- · will be treated confidentially;
- · will only be used in anonymised form for this research;
- · will be stored so that unauthorized persons cannot access them.

If you have questions about your rights as a research participant or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher, please contact the Secretary of the Ethics Committee of the department of EEMCS, mail: <u>ethics-comm-ewi@utwente.nl</u>

Thank you for taking part in this research!

Yours sincerely,

Iryna Kulatska <u>i.kulatska@student.utwente.nl</u>

\* Required

#### By choosing "I consent" option, you confirm you hereby understood your rights and agree to continue. \*

- I consent
- I do not consent

NEXT

Lleen.		
User	Exper	ience

Please write down the identification code, that can be found the top of the ArgueBot webpage.	on
Your answer	
Was the information about the debates clearly presented? *	
O Yes	
O No	
Please elaborate on your answer above	
Your answer	
Did you find the information about the debates helpful? *	
O Yes	
O №	
-	
Please elaborate on your answer above	
Your answer	
how natural(human-like) you perceived it. * 1 2 3 4 5 6 7 8 9 10	,
Unnatural O O O O O O O O O Nat	ural
Please elaborate on your answer above	
Your answer	
How did you perceive the grammar used by the chatbot? *	
1 2 3 4 5 6 7 8 9 10	
very bad O O O O O O O O O very g	ood
Please elaborate on your answer above	
Your answer	
How would you rate the response quality of the chatbot. $^{\star}$	
1 2 3 4 5 6 7 8 9 10	
Unsatisfactory	tory
Please elaborate on your answer above	
Your answer	
Please give any additional feedback here	
Your answer	
BACK SUBMIT	

Never submit passwords through Google Forms.

### Appendix Survey ArgueBot 2.0

# B

#### ArgueBot 2.0

This research is a part of master thesis, entitled "ArgueBot: Enabling a debate through a hybrid retrieval-generation-based chatbot". The aim of the research is to create a chatbot that is able to maintain a meaningful debate with users on various topics throughout the whole interaction. The goal of the chatbot is to inform, challenge and provoke the user into critical thinking.

The data collected by the chatbot and this survey:

- · will not be passed on to third parties;
- · will be treated confidentially;
- · will only be used in anonymised form for this research;
- · will be stored so that unauthorized persons cannot access them.

If you have questions about your rights as a research participant or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher, please contact the Secretary of the Ethics Committee of the department of EEMCS, mail: <u>ethics-comm-ewi@utwente.nl</u>

Thank you for taking part in this research!

Yours sincerely,

Iryna Kulatska i.kulatska@student.utwente.nl

\* Required

## By choosing "I consent" option, you confirm you hereby understood your rights and agree to continue. \*

- I consent
- I do not consent

#### NEXT

Never submit passwords through Google Forms.

	ce
Please write of the top of the	own the identification code, that can be found on ArgueBot webpage. *
Your answer	
Have you ever	had any interaction with a chathot before? *
	t version 1 and other chatbots
Yes Arquebo	t version 1 only
Yes, other ch	atbots
O No	
Was the infor	nation about the debates clearly presented? *
() Yes	
O No	
Please elabor	ate on your answer above
Did you find th background o O Yes	e information about the debates helpful (the n the left side of the chatbot)? *
() No	
Please elabor	ate on your answer above
Vour answer	
Please rate th	e conversation flow with the chatbot in terms of uman-like) you perceived it. *
Please rate th how natural(h Unnatural	e conversation flow with the chatbot in terms of uman-like) you perceived it. * 2 3 4 5 6 7 8 9 10 0 0 0 0 0 0 0 0 0 0 0 Natural
Please rate th how natural(h Unnatural C Please elabor	e conversation flow with the chatbot in terms of uman-like) you perceived it. * 2 3 4 5 6 7 8 9 10 0 0 0 0 0 0 0 0 0 0 0 0 0 Natural ate on your answer above
Please rate th how natural(h Unnatural C Please elabor Your answer	e conversation flow with the chatbot in terms of uman-like) you perceived it. * 2 3 4 5 6 7 8 9 10 0 0 0 0 0 0 0 0 0 0 0 Natural ate on your answer above
Please rate th how natural (h Unnatural C Please elabor Your answer How did you p 1 very bad O	e conversation flow with the chatbot in terms of uman-like) you perceived it. * 2 3 4 5 6 7 8 9 10 0 0 0 0 0 0 0 0 0 0 0 0 0 Natural ate on your answer above erceive the grammar used by the chatbot? * 2 3 4 5 6 7 8 9 10 0 0 0 0 0 0 0 0 0 0 0 0 very good
Please rate th how natural (h Unnatural C Please elabor Your answer How did you p 1 very bad O Please elabor Your answer	e conversation flow with the chatbot in terms of uman-like) you perceived it. * 2 3 4 5 6 7 8 9 10 Natural ate on your answer above Perceive the grammar used by the chatbot? * 2 3 4 5 6 7 8 9 10 Very good ate on your answer above
Please rate th how natural(h Unnatural C Please elabor Your answer How did you p 1 very bad O Please elabor Your answer If you receive end, how wou	e conversation flow with the chatbot in terms of uman-like) you perceived it. * 2 3 4 5 6 7 8 9 10 Natural ate on your answer above Perceive the grammar used by the chatbot? * 2 3 4 5 6 7 8 9 10 very good ate on your answer above damagenerated sentences marked with 'GR' at the ld you rate their response quality? 1 2 3 4 5 6 7 8 9 10
Please rate th how natural (h Unnatural C Please elabor Your answer How did you p 1 very bad O Please elabor Your answer If you receive end, how wou	e conversation flow with the chatbot in terms of uman-like) you perceived it. * 2 3 4 5 6 7 8 9 10 Natural ate on your answer above Perceive the grammar used by the chatbot? * 2 3 4 5 6 7 8 9 10 very good ate on your answer above Chang generated sentences marked with 'GR' at the id you rate their response quality? 1 2 3 4 5 6 7 8 9 10 Satisfactory
Please rate th how natural (h Unnatural C Please elabor Your answer How did you p 1 very bad O Please elabor Your answer If you receive end, how wou Unsatisfactory	e conversation flow with the chatbot in terms of uman-like) you perceived it. * 2 3 4 5 6 7 8 9 10 Natural ate on your answer above Perceive the grammar used by the chatbot? * 2 3 4 5 6 7 8 9 10 very good ate on your answer above damagenerated sentences marked with 'GR' at the Id you rate their response quality? 1 2 3 4 5 6 7 8 9 10 Satisfactory
Please rate th how natural (h Unnatural C Please elabor Your answer How did you p 1 very bad O Please elabor Your answer If you receive end, how wou Unsatisfactory Please elabor Your answer	e conversation flow with the chatbot in terms of uman-like) you perceived it. * 2 3 4 5 6 7 8 9 10 Natural ate on your answer above Perceive the grammar used by the chatbot? * 2 3 4 5 6 7 8 9 10 very good ate on your answer above A state on your answer above
Please rate th how natural (h Unnatural C Please elabor Your answer How did you p 1 very bad O Please elabor Your answer If you receive end, how would Unsatisfactory Please elabor Your answer	e conversation flow with the chatbot in terms of uman-like) you perceived it. * 2 3 4 5 6 7 8 9 10 Natural ate on your answer above Perceive the grammar used by the chatbot? * 2 3 4 5 6 7 8 9 10 very good ate on your answer above damy generated sentences marked with 'GR' at the ld you rate their response quality? 1 2 3 4 5 6 7 8 9 10 Satisfactory ate on your answer above
Please rate th how natural (h Unnatural C Please elabor Your answer How did you p 1 very bad O Please elabor Your answer If you receive end, how wou Unsatisfactory Please elabor Your answer How would yo sentences in	e conversation flow with the chatbot in terms of uman-like) you perceived it. * 2 3 4 5 6 7 8 9 10 Natural ate on your answer above Perceive the grammar used by the chatbot? * 2 3 4 5 6 7 8 9 10 very good ate on your answer above dany generated sentences marked with 'GR' at the Id you rate their response quality? 1 2 3 4 5 6 7 8 9 10 Satisfactory ate on your answer above
Please rate th how natural (h Unnatural C Please elabor Your answer How did you p 1 very bad O Please elabor Your answer If you receive end, how wou Unsatisfactory Please elabor Your answer How would you	e conversation flow with the chatbot in terms of uman-like) you perceived it. * 2 3 4 5 6 7 8 9 10 Natural ate on your answer above Perceive the grammar used by the chatbot? * 2 3 4 5 6 7 8 9 10 very good ate on your answer above dany generated sentences marked with 'GR' at the ld you rate their response quality? 1 2 3 4 5 6 7 8 9 10 Satisfactory ate on your answer above United the chatbot? (not marked with 'GR' at the end) * 1 2 3 4 5 6 7 8 9 10 Satisfactory ate on your answer above
Please rate th how natural (h Unnatural C Please elabor Your answer How did you j 1 very bad O Please elabor Your answer If you receive end, how wou Unsatisfactory Please elabor Your answer How would yy sentences in	a 3 4 5 6 7 8 9 10     2 3 4 5 6 7 8 9 10     2 3 4 5 6 7 8 9 10     ate on your answer above      Preceive the grammar used by the chatbot?*   2 3 4 5 6 7 8 9 10   Natural   ate on your answer above   Preceive the grammar used by the chatbot?*   2 3 4 5 6 7 8 9 10   Very good ate on your answer above Any generated sentences marked with 'GR' at the ld you rate their response quality?   1 2 3 4 5 6 7 8 9 10   Any generated sentences marked with 'GR' at the eld you rate their response quality?   1 2 3 4 5 6 7 8 9 10   Any generated sentences marked with 'GR' at the eld you rate their response quality?   1 2 3 4 5 6 7 8 9 10   Any generated with 'GR' at the end) *   1 2 3 4 5 6 7 8 9 10   Any generated with 'GR' at the end) *   1 2 3 4 5 6 7 8 9 10   Any generated with 'GR' at the end) *   1 2 3 4 5 <t< td=""></t<>
Please rate th how natural (h Unnatural C Please elabor Your answer How did you p 1 very bad O Please elabor Your answer If you receive end, how wou Unsatisfactory Please elabor Your answer How would yu sentences in Unsatisfactory Please elabor	e conversation flow with the chatbot in terms of uman-like) you perceived it. * 2 3 4 5 6 7 8 9 10 Natural ate on your answer above Perceive the grammar used by the chatbot? * 2 3 4 5 6 7 8 9 10 very good ate on your answer above date on your answer above ate on your answer above date on your answer above ate on your answer above A 1 2 3 4 5 6 7 8 9 10 Natural A 5 7 8 9 10 Natural A 5 8 8 9 10 Natural A

Your answer

BACK SUBMIT